MANCHESTER 1824
The University of Manchester

# Factualizing Biomedical Event Extraction

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Science and Engineering

2024

Jake A Vasilakes
Department of Computer Science

# Contents

**Contents** 2

**List of figures** 5

**List of tables** 9

**List of publications** 13

**Abstract** 14

**Declaration of originality** 15

**Copyright statement** 16

**Acknowledgements** 17

**1 Introduction** 18
- 1.1 From Facts to Beliefs . . . 20
- 1.2 Motivation for the Focus on Biomedical Literature . . . 24
- 1.3 Research Aims and Thesis Structure . . . 24

**2 Background** 26
- 2.1 Linguistic Preliminaries . . . 26
- 2.2 Event Extraction . . . 27
  - 2.2.1 Methods for Event Extraction . . . 28
  - 2.2.2 Takeaways . . . 38
- 2.3 Context Classification . . . 38
  - 2.3.1 Negation and Speculation Processing . . . 39
  - 2.3.2 Datasets . . . 42
- 2.4 Conclusion . . . 43

**3 SemMed Context: A large dataset of biomedical events with negation and speculation annotations** 45
- 3.1 Background . . . 46
  - 3.1.1 Weak Supervision . . . 46
  - 3.1.2 SemRep and SemMedDB . . . 49
  - 3.1.3 SemRep Factuality . . . 49
- 3.2 SemMed Context . . . 51
  - 3.2.1 Construction . . . 51

3.2.2 Evaluation of the filtering pipeline . . . 54
3.2.3 Dataset Description . . . 54
3.3 Baseline Model . . . 55
3.3.1 Model Description and Evaluation Setup . . . 55
3.3.2 Results . . . 55
3.4 Computing Consensus over Claims . . . 57
3.4.1 Background: Subjective Logic . . . 57
3.4.2 SemMed Context as a Knowledge Graph . . . 59
3.4.3 Preliminary Work: Embracing Disagreement . . . 61
3.5 Limitations and Future Work . . . 62

**4 Disentangling Event Content from Polarity and Uncertainty 64**
4.1 Background . . . 65
4.1.1 Variational AutoEncoders . . . 65
4.1.2 Disentangled Representation Learning . . . 66
4.1.3 Contributions . . . 67
4.2 Generative Model . . . 68
4.3 Desiderata for Disentanglement . . . 70
4.3.1 Informativeness . . . 70
4.3.2 Independence and Invariance . . . 70
4.4 Experiments . . . 72
4.4.1 Datasets . . . 72
4.4.2 Evaluation . . . 74
4.5 Implementation Details . . . 75
4.6 Results . . . 76
4.6.1 Disentanglement . . . 76
4.6.2 Evaluation of Reconstructions . . . 78
4.6.3 Discussion . . . 81
4.7 Disentangled Representation Learning on SemMed Context . . . 82
4.7.1 BART as a Variational Autoencoder . . . 83
4.7.2 SubjVAE . . . 85
4.7.3 FactVAE . . . 87
4.7.4 Evaluation . . . 88
4.7.5 Discussion . . . 92
4.7.6 Limitations and Future Work . . . 94
4.8 Conclusion . . . 95

**5 Factualized Event Extraction with Interpretable Predictions 96**
5.1 Levitated Context Markers . . . 97
5.1.1 Background . . . 98
5.1.2 Data . . . 99
5.1.3 Model . . . 100
5.1.4 Results . . . 102

5.1.5 Discussion . . . 105
5.1.6 Limitations and Future Work . . . 106
5.1.7 Implementation Details . . . 106
5.2 Levitated Context Markers for Multi-Task FEE . . . 107
5.2.1 Extended Model . . . 107
5.2.2 Preliminary Experiments . . . 109
5.2.3 Improving Multi-Task Performance on Cue Detection . . . 110
5.2.4 Results on Test Splits . . . 114
5.3 Conclusion . . . 115

**6 Conclusion** **117**
6.1 Summary of Results . . . 117
6.2 Future Work . . . 118
6.2.1 Data . . . 119
6.2.2 Models . . . 119
6.3 Large Language Models . . . 121
6.4 Final Thoughts . . . 122

**References** **123**

**Appendices** **160**

**A Subjective Logic Encodings** **161**

**B Software** **172**

**C Evaluating LLMs for Classifying Polarity and Uncertainty** **182**

**Word count**: 38,298

# List of figures

1.1 Example atomic propositions, complex propositions, and the corresponding Euler diagram with sections colored accordingly. The outermost portion $S$ is the set of all possible situations and the inner portions describe possible worlds according to the complex propositions. The point $w$ is one of these possible worlds. . . . . 21
1.2 Barycentric triangle representation of factuality, with axes for polarity and uncertainty. . . . . 23

2.1 A high-level overview of an EE system. The input text is preprocessed (tokenization, normalization) and sent to a named-entity recognition (NER) module which identifies all the entities. The EE module predicts the event structure according to the representation schema. In this example, which uses a <*subject, verb, object*> schema, the EE module would predict which of each pair of entities is the *subject* and *object*, respectively, as well as the *verb* that connects them. . . . . 28
2.2 A timeline of research milestones in event extraction, split up into the three major periods: rule-based/machine learning, deep neural networks, and large pre-trained language models. The current state-of-the-art methods for relation extraction (RE) and event extraction (EE) are indicated in red. . . . . 29
2.3 A simplified example of how an attention mechanism works for relation extraction. Given the sentence and a target relation triple, the attention mechanism learns weights for each token such that when added together they are a good predictor of the target. Higher attention weights are given darker orange highlights. 33
2.4 A depiction of the vanishing gradient issue in RNNs. The sentence representation is computed as the last hidden state $h_{10}$ of the input sequence. Given the vanishing gradient, one of the most important input words *dog* has little effect on the final representation. . . . . 34
2.5 A comparison of the token embedding spaces of non-contextual models such as Word2Vec to contextual models such as BERT. Word2Vec is a simple key-value table of tokens to embeddings, meaning that homographs such as *bow* are encoded the same despite their very different meanings. BERT, on the other hand, computes token embeddings dynamically according to the other tokens in the input. This means that homographs such as *bow* can be encoded differently. Note that *refuse* is also a homograph, but only one of its meanings is used here so it is always encoded the same. . . . . 35
2.6 Example of relation triple linearization. Reproduced from [111]. . . . . 38

2.7 The GENIA example from Figure 2.9c with the addition of the negation cue. . 39
2.8 A timeline of research milestones in negation and speculation processing, split into periods akin to those in Figure 2.2: rule-based/machine learning, deep neural networks, and large pre-trained language models. The current state-of-the-art methods for scope resolution and event context classification are indicated in red. . . . . 40
2.9 Example annotations from the CMED, SemRep, and GENIA datasets. . . . . 44

3.1 An overview of the SemRep pipeline. . . . . 49
3.2 An overview of the SemMed Context construction process, broken up into 5 steps. 51
3.3 Examples of aleatoric $\oplus$ and epistemic $\ddot{\oplus}$ cumulative fusion of two subjective opinions depicted as Beta PDFs. . . . . 59
3.4 Representation of the factuality values in SemRep Factuality as Beta distributions. The negation dimension is represented by the $x$-axis and the certainty dimension by the distribution variance. The mapping of factuality values to scalars is from [39]. Note that while the Fact and Counterfact distributions have 0 variance, we add variance to the figure for ease of visualization. . . . . 60
3.5 A subgraph of SemMed Context showing the relationships between Epstein-Barr Virus, three diseases, and their respective treatments, all with $\geq 2$ supporting claims. The consensus belief of each relationship is represented by the darkness of the arrow color and we label relationships with less than a Fact label with their corresponding belief value in parentheses. . . . . 61

4.1 Example indicating the distinction between polarity, uncertainty, and content. The content "Trees have leaves" is modified by the negation (**bold**) and speculation (underlined) expressions. . . . . 64
4.2 Graph of the basic generative model estimated by a VAE. $X$ is the observed data and $Z$ is the unobserved latent. . . . . 65
4.3 A diagram of a Variational AutoEncoder. The input data $x$ is encoded by a neural network $\phi$ into a latent distribution. The distribution is sampled to obtain a latent value $z$, which is decoded by another neural network $\theta$ that attempts to reconstruct the input. . . . . 66
4.4 Graph of the generative model. $c$ is content, $n$ is polarity (negation status), and $u$ is uncertainty. . . . . 68
4.5 The proposed architecture corresponding to the $\mathcal{L}_{ELBO} + \mathcal{L}_{INF}$ objective (see Section 4.3.2). A BiLSTM encoder parameterized by $\phi$ maps each input example $x$ to three distinct distributions from which the latent representations $z^{(\ell)}$ are sampled. The negation $z^{(n)}$ and uncertainty $z^{(u)}$ latent spaces are then passed to linear classifiers, parameterized by $\psi^{(\ell)}$, which attempt to predict the ground truth factor. Finally, the latent values initialize an LSTM decoder, parameterized by $\theta$, which attempts to reconstruct the input. . . . . 69
4.6 Histogram and t-SNE visualizations of the polarity (negation), uncertainty, and content spaces learned by the INF+ADV+MIN model. . . . . 76

4.7 Box plots of the Mutual Information Gap (MIG) for each disentanglement objective for the negation and uncertainty factors computed on the test sets. The MIG values for the baseline $\mathcal{L}_{\text{ELBO}}$ objective were too small to include in this figure. . . . . 77
4.8 Graph of the generative model. $n$ is polarity (negation) status, $u$ is uncertainty, $p$ is propositional content, and $s$ is style. . . . . 83
4.9 An illustration of SubjVAE. The BART encoder outputs a latent distribution for each task. For SemMed Context these are Betas for polarity and uncertainty (uncertainty not shown for space reasons), an SL Dirichlet for predicate type, and a multivariate Gaussian for the text style. As in a standard VAE, these distributions are sampled using a deterministic sampling method to obtain latent values $z$. The latent values are input to the BART decoder, which predicts the next token from the preceding (e.g., $x_1$ from <s>) with cross attention between the preceding tokens and the latent values. . . . . 86
4.10 Illustration of a Dirichlet prior with samples corresponding to Beta distributions. 87
4.11 An illustration of FactVAE. The BART encoder outputs a 3-dimensional Dirichlet, which is sampled to obtain the $b, d, u$ parameters of an SL Beta PDF representing the event factuality, which is used to predict the polarity and uncertainty labels. The encoder also outputs a Gaussian distributions for style and SLBetas or SLDirichlets for any other discrete generative factors (e.g., predicate type is modeled using an SLDirichlet as in SubjVAE, but is not shown for space reasons), which are sampled to obtain latent vectors $\boldsymbol{z}$. The latent values $b, d, u$, and $\boldsymbol{z}$ are input to the BART decoder, which predicts the next token from the preceding (e.g., $x_1$ from <s>) with cross attention between the preceding tokens and the latent values. . . . . 88
4.12 Illustration of a Dirichlet prior with samples corresponding to Beta distributions. 93
4.13 Example distributions predicted by FactVAE for CounterFact, Doubtful, Possible, and Fact events. . . . . 93

5.1 An illustration of our proposed model. An input sentence is passed to three different heads for predicting the event, negation, and uncertainty, respectively. Each head makes its prediction based only off of the relevant subspans in the input. For example, the Negation head would make its prediction using the two entity spans *"Trees"* and *"leaves"* as well as the negation cue *"not"*. . . . . 97
5.2 Comparison of levitated markers for their original use case of relationship extraction (A) and our task of context classification (B). In (A), a hypothetical relationship extraction task, the subject of the relation is given solid markers ([S] , [/S]) and all candidate objects are given levitated markers (e.g., L1, with colors indicating shared position IDs). A classifier then makes a prediction for each subject-object pair. In (B), the medication span is given solid markers and multiple spans are given levitated markers to act as the context. Instead of making a prediction for each pair, the levitated span representations are pooled and used to make a single prediction. . . . . 99

5.3 Example cues detected by LCMs with sparse attention pooling on each context dimension. Examples of Relevant cues from each context dimension. The **medication** is in orange and **cues** are highlighted in green proportional to their attention weight. Gold labels are provided where necessary to aid interpretation of the cues. . . . . 105

5.4 An illustration of the hierarchical levitated context marker model. A shared pretrained language model (PLM) encodes the input tokens $\boldsymbol{x}$ into hidden representations $\boldsymbol{h}$. Then, task-specific layers further encode the hidden representations and pass them to levitated context marker (LCM) layers. In this hierarchical model, the output $\boldsymbol{z}$ of the LCM layer for the predicate prediction task (in **blue**) is input to the LCM layer for the auxiliary tasks (in **green**). The **red** dashed arrows indicate the path of gradient back-propagation. Note that gradients from the auxiliary tasks are not propagated through $\boldsymbol{z}$ into the layers meant for the predicate task, in order to keep the task-specific layers independent. 111

5.5 Sparse attentions from the Hierarchical Projected Attention model on the polarity, uncertainty, and predicate tasks showing the detected cues. **Subject** entities are in blue, **objects** in orange, and **cue spans** are highlighted in green proportional to their attention weight. . . . . 114

5.6 A negated and uncertain PREVENTS claim highlighted with the highest attention-weighted cue spans for each context dimension, colored as follows: **polarity**, **uncertainty**, **predicate**. The **subject** entity is blue and the **object** is orange. . . . . 114

A.1 F1, JSD, and NES results for Majority Voting (MV), Soft voting (Soft), and SLE cumulative fusion (Fused) given both all and filtered annotations on the synthetic data across different ranges of uncertainty. Results using CrowdTruth were nearly identical to Soft, and so omitted here for clarity. . . . . 166

# List of tables

2.1 Summary statistics of negation and speculation annotations in English biomedical- and general-domain corpora, ordered from largest to smallest. . . . . 43

3.1 The mapping from the factuality values in SemRep Factuality to polarity and uncertainty values. . . . . 50
3.2 Performance of the rule-based factuality classifier introduced in [39] on their gold-standard test set, as well as the precision of each label manually evaluated on a separate held-out set, randomly selected from SemMedDB and entirely distinct from the gold-standard test set. . . . . 50
3.3 Number and percentage of each predicate extracted from SemMedDB. . . . . 52
3.4 Number and percentage of each predicate extracted from SemMedDB after filtering. . . . . 52
3.5 Number and percentage of each predicate extracted from SemMedDB after filtering, along with the number of Negated and Uncertain examples for each. The percentages for each predicate are computed relative to the total number of examples, while the percentages of Negated and Uncertain examples are computed relative to the number of the corresponding predicate. . . . . 53
3.6 Number and percentage of each predicate extracted from SemMedDB after negation filtering, along with the number of Negated and Uncertain examples for each. The percentages for each predicate are computed relative to the total number of examples, while the percentages of Negated and Uncertain examples are computed relative to the number of the corresponding predicate. . . . . 54
3.7 Precision, Recall, and F1 score of SemRep Factuality on the gold-standard test set. Precision of each negation, and uncertainty, and factuality label assigned by SemRep Factuality before (P) and after (P Filtered) filtering, compared to the gold-standard results. . . . . 54
3.8 Comparison of the SemRep Factuality and SemMed Context splits. Note that SemRep Factuality does not have a validation split. We omit the Conditional and Uncommitted Factuality examples from SemRep Factuality due to low counts and not having polarity/uncertainty labels, respectively. . . . . 55
3.9 Results of the baseline BERT model trained on SemMed Context and evaluated on both SemMed Context and the SemRep Factuality test set. For the predicate task the model was evaluated on the subset of the SemRep Factuality test set that contained the predicates used in SemMed Context. For the polarity model, we weight the loss function according to the proportion of classes in the training data to account for label imbalance. . . . . 56

3.10 The baseline BERT models compared to the best performing model from [39] on the SemRep Factuality test set. BERT Factuality: The baseline model fine-tuned on Factuality annotations. BERT Neg+Cert: The baseline model fine-tuned on Negation and Certainty annotations, using the equivalences in Table 3.1 to map predictions back to Factuality. . . . . 56
3.11 Results of the baseline BERT model trained on SemRep Factuality and evaluated on both SemRep Factuality and the SemMed Context test set. Note that for model development we chose a stratified subset of examples from the SemRep Factuality train set to use for validation. We also only trained and evaluated the model on the subset of SemRep Factuality that contained the predicates used in SemMed Context. For the Negation model, we weight the loss function according to the proportion of classes in the training data to account for label imbalance. There are too few examples in the train set to estimate a Predicate model, so we do not report those results here. . . . . 57
3.12 Classification of subjective opinions in Subjective Logic according to the values of the $b$, $d$, and $u$ parameters. . . . . 58
3.13 Count of each Factuality value in SemMed Context after consensus, computed using thresholds of the belief value $b$. . . . . 60

4.1 The tokens used as features for the negation and speculation classifiers. Contractions are split in our tokenization scheme. . . . . 73
4.2 Precision, recall, and F1 score of the negation and uncertainty classifiers on the SFU dev and test sets. . . . . 74
4.3 Statistics for the combined SFU and weakly labeled Amazon and Yelp datasets. 74
4.4 Informativeness results on SFU+Amazon and SFU+Yelp showing the Mutual Information estimate between each latent and factor as well as Precision, Recall, and F1 of the latent space classifiers for each factor. $n$: polarity (negation). $u$: uncertainty. $c$: content. Shown are the mean values computed from 30 resamples of the latent distributions for each example. . . . . 76
4.5 Pearson's correlation coefficients between each pair of latent representations across models on SFU+Yelp. As the content space is 62-dimensional, we compute the correlation coefficient between each dimension and report the mean and the standard deviation. . . . . 78
4.6 Reconstruction self-BLEU and reconstruction perplexity (PPL) for each model on each data split. Perplexity is computed using GPT-2 [108]. . . . . 79
4.7 Example reconstructions decoded by the INF+ADV+MI model on the test sets. 80
4.8 Consistency of the decoder with the ground-truth values of negation and uncertainty evaluated on the test set. Pass 1 refers to the predictions from the original inputs. Pass 2 refers to the predictions from the re-encoded reconstructions. Pass 1 can be considered an upper bound on the performance of pass 2. . . . . 80
4.9 Attribute inversion accuracies by factor and direction of transfer using the $\mathcal{L}_{\text{INF}}$+$\mathcal{L}_{\text{ADV}}$+$\mathcal{L}_{\text{MIN}}$ model on the test sets. . . . . 80

4.10 $R^2$ of the sentence length regression models on $z$'s sampled from INF+ADV+MIN model trained on SFU+Amazon. $R^2 = 1$ represents perfect predictive ability. . 80
4.11 Successful and Unsuccessful inversions of polarity and uncertainty. Inputs are in standard font and reconstructions are in *italics*. **bold** tokens indicate negation/speculation cues. . . . . 81
4.12 Informativeness results of SubjVAE on the validation set, including the mutual information (MI) between the latent and the ground-truth labels and the macro-averaged (P)recison, (R)ecall, and F1 score. . . . . 89
4.13 Informativeness results of FactVAE on the validation set, including the mutual information (MI) between the latent and the ground-truth labels and the macro-averaged (P)recison, (R)ecall, and F1 score. . . . . 89
4.14 The Mutual Information Gap (MIG) for SubjVAE and FactVAE for each generative factor. Note that the MIG for polarity and uncertainty are both computed from the factuality latent. The reported results are averaged over 10 runs. . . . 89
4.15 Invariance results on the validation set, measured as the Pearson's correlation between samples from each latent space. . . . . 90
4.16 Invariance results on the dev set. . . . . 90
4.17 Reconstruction (BLEU) and perplexity (PPL) of the text generated on each data split for SubjVAE and FactVAE. Perplexity is computed using GPT-Neo [218] fine-tuned on scientific text. For comparison with the reconstructions, PPL (inputs) is the perplexity of the gold-standard input text computed under GPT-Neo. 90
4.18 Example reconstructions decoded by SubjVAE on the validation set. . . . . 91
4.19 Example reconstructions decoded by FactVAE on the validation set. . . . . 92
4.20 Results of SubjVAE and FactVAE on the SemMed Context test split, compared to the baseline BERT model from Section 3.3. . . . . 92

5.1 Context dimensions along with their definitions and possible labels. . . . . 100
5.2 Dataset statistics for the train, validation, and test sets for the Event Extraction and Context Classification tasks. . . . . 100
5.3 Micro and Macro averaged (P)recision, (R)ecall, and F1 score of event detection on the validation set of CMED. **Baseline**: No levitated markers; solid markers surrounding the medication span only. **LCMs**: Solid markers and max pooled LCMs. **+types**: Typed levitated markers according to the semantic tags of the verbs. The best numbers in each column are in **bold**. . . . . 103
5.4 Macro averaged (P)recision, (R)ecall, and F1 score of context classification on CMED for each pooling method and context dimension. The best score for each metric is in **bold**. The model for each dimension with the best F1 score on the validation set was evaluated on the test set given gold standard event input (Gold) and predictions from the best EE model (E2E). **Baseline**: No levitated markers; solid markers surrounding the event span only. **Maxpool**: Max pooling of the levitated marker representations. **Softmax**: Attention pooling with softmax projection. **Sparse**: Attention pooling with Sparsegen-lin projection. . 104

5.5 Accuracy of the cues detected by the sparse attention pooling mechanism, computed as the percentage of Relevant cues. . . . . 104
5.6 Hyperparameter settings for the Event Extraction models. . . . . 107
5.7 Hyperparameter settings for the Context Classification models. $W$: window size parameter for the Levitated Context Markers. $\lambda$: sparsity coefficient for the Sparsegen-lin projection function. . . . . 107
5.8 (P)recision, (R)ecall, and F1 score for the single-task models on the Polarity, Certainty, and Predicate tasks, respectively. The best F1 score for each task is in **bold**. N.B., the results for the CLS model are the same as those reported in Table 3.9 in Chapter 3. . . . . 108
5.9 Results for the multi-task models jointly trained for 5 epochs on the Polarity, Certainty, and Predicate tasks. F1 scores higher than the best single-task model from Table 5.8 are in **bold**. . . . . 109
5.10 Percentage of correct predictions with relevant cues from the sparse LCM model on the validation set. . . . . 110
5.11 Prediction performance for the flat vs hierarchical sparse LCM models jointly trained on the polarity, uncertainty, and predicate type tasks. . . . . 112
5.12 Results of the sparse attention models on the cue detection task, including the percentage of predictions with relevant cues as well as average input token coverage of the sparse attention masks. Flat-ST refers to the single-task models with sparse attention from Table 5.8. . . . . 113
5.13 Results of the baseline BERT model on both SemMed Context and the SemRep Factuality test set. For predicate the model was evaluated on the subset of the SemRep Factuality test set that contained the predicates used in SemMed Context. 115
5.14 Cue detection results of the hierarchical projected attention model on the SemRep Context and SemRep Factuality test splits. . . . . 115

6.1 Macro-averaged results of the baseline BERT model, FactVAE, and the best LCM model variant (Hierarchical with PAL layers) on the SemMed Context test split. . . . . 118

A.1 A comparison of methods for learning from annotation uncertainties. . . . . 170
A.2 Hard and soft metric results on the real-world data sets. CIFAR-10S is the only real-world dataset that provides both annotator confidence and inter-annotator disagreement. ConvAbuse does not contain gold-standard labels. * MTLSL is trained using both gold and crowd annotations. . . . . 170

C.1 Results of Mixtral 8x7B Instruct and Gemma 2 9B Instruct evaluated on subsets of 100 examples from the SemMed Context validation set, stratified to contain equal examples of each label for the Polarity and Uncertainty tasks, respectively. Predictions from each model were obtained zero-shot. . . . . 183

# List of publications

J. Vasilakes, C. Zerva, M. Miwa, and S. Ananiadou, "Learning disentangled representations of negation and uncertainty," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8380–8397. doi: 10.18653/v1/2022.acl-long.574. [Online]. Available: https://aclanthology.org/2022.acl-long.574.

J. Vasilakes, P. Georgiadis, N. T. H. Nguyen, M. Miwa, and S. Ananiadou, "Contextualized medication event extraction with levitated markers," Journal of Biomedical Informatics, vol. 141, p. 104 347, May 2023, issn: 1532-0464. doi: 10.1016/j.jbi.2023.104347.

# Abstract

Recent research describes a replication crisis in science due to an overwhelming amount of scientific literature and a corresponding decline in quality. Event extraction (EE), a subtask of natural language processing (NLP), has the potential to mitigate the ensuing information overload by automatically extracting structured knowledge representations and making scientific literature easily searchable. However, existing methods for EE focus on representing only the core aspects of events –who did what to whom– which constitutes an assumption that all extracted events are *true*. We argue that this assumption is fundamentally problematic because it dooms EE systems to misrepresent the world, especially in the scientific domain where the claims are often expressed as false or uncertain.

To address this problematic assumption, we propose a fundamental conceptual shift in EE from the representation of facts to the representation of *beliefs*. While a fact is necessarily (assumed to be) true, beliefs lie on a continuous scale of truth and uncertainty, which together constitute the notion of *factuality*. We thus propose that EE systems include *de facto* dimensions of polarity and certainty in addition to the core representation.

The importance of these dimensions to EE is far from a new realization, but to our knowledge the necessity of fundamentally integrating factuality into the core event representation has not yet been established in NLP research. One contribution of this thesis is theoretical arguments for this integration. Another contribution is the introduction of the task of Factualized Event Extraction (FEE), which targets the joint extraction of events' core representation alongside the expressed polarity and uncertainty. As part of this, we introduce SemMed Context, a new dataset for FEE in the biomedical domain, and describe the development of multi-task models for FEE based on variational autoencoders (VAEs) and pre-trained language models (PLMs). Experiments with both models show a large improvement over a baseline, and even suggest that multi-task learning of event content, polarity, and uncertainty performs as well as or better than learning separate models for each. In addition to good prediction performance, we introduce methods that lend interpretability to the model predictions: interpretable latent distributions in the VAE and an interpretable attention mechanism in the PLM-based model.

# Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright statement

i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.library.manchester.ac.uk/about/regulations/`) and in The University's policy on Presentation of Theses.

# Acknowledgements

In the first instance I must thank my supervisor Professor Sophia Ananiadou, whose support and dedication allowed me to pursue what interested me at a pace that suited me. This was especially appreciated in the early stages of the PhD where I was working remotely in the US due to COVID restrictions. I also have to thank Chrysoula Zerva, a graduate of NaCTeM whose work directly inspired much of my research and who helped make my first ACL publication and Chapter 4 of this thesis possible. It was wonderful working and chatting with her, even though we never managed to meet in person. I am also indebted to Panagiotis Georgiadis and Nhung Nguyen –excellent research partners and friends– whose insights led to the development of LCMs and Chapter 5.

My wife Hannah has perhaps been the most important person throughout this process. She has supported and encouraged me from the first inkling of wanting to do a PhD through to the end even though it meant moving to a new country, working long and strange hours, and taking a severe pay cut. Her love and understanding made completing the PhD possible. She also brought our two wonderful cats in my life, Mischa and Hamish, whose periodic demands for attention, food, and play were welcome reminders to take a break from my work and focus on the simpler things.

# Chapter 1

# Introduction

*And how will you enquire, Socrates,*
*into that which you do not know?*
*What will you put forth as the subject of enquiry?*
*And if you find what you want,*
*how will you ever know*
*that this is the thing which you did not know?*

— Plato, *Meno*, 80d

I often take walks around my neighborhood. My route necessarily starts with what I know: out my flat, down the lift, through the building lobby, out the door, and down the street to the path. Soon, however, I am faced with paths and streets I have not taken before, branching out from my known route and towards different places I might visit. Shops, gardens, plazas, and alleyways await me depending on the path I choose. On a whim I start down one, and I am soon faced with a part of the city I have not seen before. When I first moved to Manchester I could discover whole neighborhoods, but after a few hundred walks from my flat I am now content to encounter a new side street or shortcut. Still, as long as I am open to the possibilities of new paths, the parts of the city I do know will continue to guide me to parts I have not yet discovered.

Whether going for a walk, talking with a friend, or writing a thesis, our lives are brimming with possibility. We are always looking out upon a metaphorical horizon of possibilities extending from and determined by our current perspective. Some of what we see on this horizon is clear and some can be inferred from past experience, but there is plenty that is far away and obscured by atmospheric haze or variations in the landscape, and it is only by venturing outward into the horizon that we can dismiss the haze or surmount the hills to expand our world. Indeed, the horizon metaphor reminds us that learning is rarely accomplished by a shot in the dark, suggesting one solution to Meno's paradox above: what we already know implies possibilities, and these possibilities are precisely the things we do not know, so it is perfectly reasonable to ask about them and learn about them. My whimsical wanderings about my neighborhood are one way of learning, but science offers another which is more rigorous, systematic, and efficient. The scientific method urges us to carefully explore the possibilities implied by what we already know, gathering information and slowly but surely expanding the totality of knowledge.

This imagery is all very idyllic, but as scientists we know that the process of discovery is in practice riddled with difficulties. We must keep exploring in the face of failed experiments, lack of funding, and a publish-or-perish culture. This last concern, along with increased specialization of scientific disciplines, has led to us being so inundated with scientific literature that it is difficult enough to figure out where we are now, much less to survey our horizon. Additionally, the pressure to publish has resulted in a general decline in quality of scientific research commensurate with its increase in volume [1]–[3]. We are left with information overload. For example, searching PubMed for information on effective therapies for Hodgkin's Lymphoma returns over 32,000 scientific articles, with an average of 700 articles published on the topic each year since 1990[1]. This is far too many papers for even a team of researchers to feasibly review, especially considering factors such as the reliability of the methods and data, date published, *in vivo* vs. *in vitro*, etc.

Event Extraction (EE), a subtask within Natural Language Processing (NLP), provides a potential solution to this information overload by automatically extracting structured data from unstructured text, making the information easily searchable and usable in downstream applications such as summarisation and automated reasoning. At a minimum, an EE system must represent the action and the agents involved. That is, it must answer the question, "Who did what to whom?". However, the "who", "what", and "whom" –hereafter referred to as the *core representation*– are only part of the picture. In order to adequately understand and use the structured information extracted by an EE system it is often important to know additional information such as "when", "where", "how", etc. All of these additional dimensions beyond the core representation we collectively refer to as the event *context*.

Context is important to EE because what is true in one context may not be true in others. For example, "The force of gravity is $9.8m/s^2$" is true on Earth, but false on Mars. This example illustrates one type of context (location), but myriad contextual aspects are potentially important to understand a knowledge claim. Indeed, the complete context –and therefore validity– of a piece of scientific knowledge is precisely the entirety of the research that supports it, including the previous research that indicated the viability of the hypothesis, the experimental methodology (e.g., sample size, testing strategies), the intermediate results, and the validity of the reasoning used for arriving at the conclusion. There has been significant work within information extraction generally to extract various contextual aspects of events (e.g., [4]–[6]), but we shall argue that two of the most fundamental within the scientific literature are polarity and uncertainty, as they are crucial to the process of knowledge acquisition. For example, a false hypothesis may rule out entire lines of inquiry, and an uncertain conclusion may warrant additional experimentation or cast doubt upon an existing theory. The polarity and uncertainty of event are ultimately determined by the event's full context (e.g., a small sample size makes a conclusion less certain, and we discuss this further in Section 3.5), but they can also manifest *per se* via their lexico-syntactic expressions, negation and speculation. Indeed, the importance of polarity and uncertainty is reflected in the frequency of negation and speculation in scientific text. Studies of the biomedical literature estimate that 13% of

[1]The query submitted was `(hodgkin lymphoma) AND (Therapy/Broad[filter])` made to `https://pubmed.ncbi.nlm.nih.gov` on 20/2/2024.

sentences in the semantic MEDLINE database are negated [7] and 11% are speculative [8].

Despite their importance, EE research has been overwhelmingly focused on the core event representation, leaving polarity and uncertainty to the wayside[2]. As evidence for this, we reviewed the schemas and annotation guidelines of publicly available EE and Relation Extraction (RE)[3] datasets listed on Papers With Code[4]. Of the 14 EE datasets[5], only 3 had both negation and speculation annotations (ACE 2005[6] [10], French Timebank [15], and Phee [19]), and only 2 others had negation only (Timebank [21] and Catalan Timebank [11]). Besides these, we found no mention of how the annotators should handle negated or speculative events in the annotation guidelines of any of the other datasets. While these 15 datasets are far from a complete list, they contain many of the most widely used EE datasets and are, we believe, indicative of the lack of attention paid to polarity and uncertainty in EE research. This oversight is not restricted to EE. Out of 13 of the most-cited RE datasets[7] we found only one –2010 i2b2 [30]– that contains negation and speculation annotations and one other –ChemProt [26]– that annotates negation. Two datasets –SemEval-2010 Task 8 [33] and SciERC [32]– explicitly instruct annotators to *ignore* negated and speculative events.

This focus on the core event representation is sensible under the assumption that the event as described in the text is true. Indeed, it is this focus on truth that motivated the authors of [33] to ignore negated and speculative events during annotation. A fundamental motivation of this thesis is that this assumption is problematic both from a theoretical and practical standpoint. Theoretically, focusing on true events is at odds with the scientific method and the process of knowledge acquisition more generally, which hinge upon investigating and ruling out possibilities. Practically, knowledge-based systems that depend on EE models which do not account for negation and speculation are doomed to incorrectly represent the world, whether they misrepresent possibilities as facts or simply ignore non-factual statements (e.g., models trained on data such as SciERC which does not annotate negated events). For these reasons, we argue that EE datasets and models ought to include dimensions of polarity and uncertainty *as part of the core event representation*, as these neatly summarise the author's belief in the truthfulness of the event. In other words, we propose to shift the subject of EE from facts to *beliefs*.

## 1.1 From Facts to Beliefs

NLP is necessarily interested in that which is said (*de dicto*) rather than the thing itself (*de re*), which means that any claims encountered by an EE system are necessarily mediated by

[2]This focus on the core representation is, we should note, more of a modern phenomenon. Early works in EE explicitly described the importance of developing methods for detection negation and speculation to the modeling of events in text (cf. [7], [9]), but this fell away in later research. A more complete picture of previous work in this area is given in Chapter 2.

[3]RE is a simplified version of the EE task with a strict event representation.

[4]`https://paperswithcode.com/`

[5]The EE datasets reviewed are, in alphabetical order: ACE 2005 [10], Catalan Timebank [11], ChFinAnn [12], EDT [13], EventNarrative [14], French Timebank [15], GENIA [16], IndiaPoliceEvents [17], M2E2 [18], Phee [19], SciREX [20], Timebank [21], Title2Event [22], and WikiEvents [23].

[6]However, most research using ACE 2005 does not attempt to predict these labels, concentrating only modeling the core event representation.

[7]The RE datasets reviewed are, in alphabetical order: ADE [24], BioRED [25], ChemProt [26], DDI [27], DocRED [28], FewRel [29], 2010 i2b2 [30], RadGraph [31], SciERC [32], SemEval 2010 Task 8 [33], T-REx [34], TACRED [35], and WebNLG [36]

$C :=$ My coat is wet.
$R :=$ It rained.
$H :=$ I washed my coat.
$D :=$ My coat was dirty.

$C \rightarrow (R \vee H)$
$D \rightarrow H$
$C \wedge D$

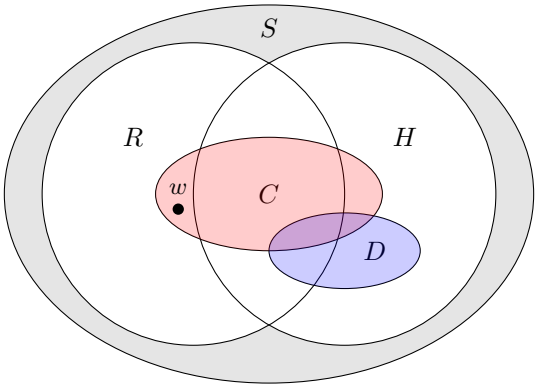



Figure 1.1. Example atomic propositions, complex propositions, and the corresponding Euler diagram with sections colored accordingly. The outermost portion $S$ is the set of all possible situations and the inner portions describe possible worlds according to the complex propositions. The point $w$ is one of these possible worlds.

the belief of the author. Despite this, NLP research has shown at best a tangential interest in the distinction between fact and belief[8]. This distinction does, however, have a rich history in philosophy and linguistics going back to the ancients (cf. Meno's paradox above and Aristotle's discussion of belief in *Posterior Analytics*, I, 2). This section therefore motivates our proposed conceptual shift from facts to beliefs via arguments from epistemology and possible-world semantics, adapted largely from [38]. These arguments show 1) that belief is the foundation of factuality and 2) that factuality is comprised of polarity and uncertainty.

We are ultimately interested in propositions, such as *"The sky is blue."* or *"TGF-$\beta$ is involved in apoptosis."*. We will define propositions as the target of the event extraction task more specifically later in Chapter 2, but generally a proposition describes some world-state. A world-state $w$ is a unique set of situations, and a situation is some particular aspect, e.g., the sky's particular shade of blue at this moment. Thus, if $S$ is the set of all possible situations, a world-state $w \subset S$ is a set of compatible situations (i.e., the set contains no contradictory situations such as *"The sky is blue."* and *"The sky is red."*). A proposition, more formally, thus describes a set of possible world-states, and a proposition is true with respect to a world-state if that world-state is a member of the set described by the proposition[9].

We can illustrate this set-theoretic definition of propositions and truth using the Euler diagram in Figure 1.1. Here, there are four atomic propositions $C, R, H$, and $D$, which comprise three complex propositions that describe possible world-states. That is, each proposition describes a set of situations (e.g., $R$ is the set particular situations in which it rained) and so each coordinate in the Euler diagram is a possible world-state[10]. The truth of a proposition can be visualized as a point (world-state) in the diagram. For example, propositions $C$ and $R$ are true in $w$ since they contain it as a member, while $H$ and $D$ are false.

If we add a temporal dimension to our premises, the Euler diagram can also help illustrate the process of knowledge acquisition. For example, of $C \wedge D$ let us assume that we first observe $C$. This implies five distinct sets of possible worlds (the shaded red and purple

[8]However, the focus on facts has been shaken in recent years with research on data perspectivism, automated fact-checking, and misinformation detection (cf. [37]).

[9]The world-state in question is often implicit. Most often it is the current, actual world, but we commonly encounter propositions that are true with respect to other possible worlds. For example, *"Samwise Gamgee is a gardener"* is true in the fictional world of Middle Earth.

[10]We can also view possible world-states as the values for which the complex propositions evaluate to *True* in the corresponding truth table.

portions of Figure 1.1), and we might say that *"My coat is wet so it may have rained or I may have washed it or both."*. We next observe $D$, which further restricts the sets of possible worlds to two (the purple shaded portion of Figure 1.1) and an expression of our deduction is "*Because my coat is wet and it was dirty, I must have washed it, although it may also have rained."*. Thus a truth in the present implies (im)possibilities in the future. Put in a more scientific context, we can see how fundamental the notions of polarity and uncertainty are to the scientific method: current truths imply (im)possibilities which are hypotheses to be tested, and each hypothesis is tested in reference to the null (negative) hypothesis.

But (im)possibility is related not only to deduction from complex propositions, as the truth of atomic propositions is rarely given *a priori*. We often come to know truth *a posteriori* through evidence –observation, measurement, or deduction– which affects our perception of truth. This points to a crucial distinction between truth and fact: as defined by [38], a fact is a true proposition that one believes to be true, and this belief is justified via reliable evidence[11]. This means that a lucky guess is not a fact because it is not justified, and even a heavily-researched and widely-accepted theory is not a fact if it is not true (e.g., geocentrism). Now, evidence is never complete, as additional evidence and more rigorous deductive reasoning will always increase the reliability of the belief. With some loose mathematical notation, we can therefore define a *fact* as the limit of belief as its evidence increases[12].

$$\lim_{r\to\infty} \text{Belief}(p, r) = \text{Fact}(p)$$

where $r \in \mathbb{R}_0^+$ is the amount of reliable evidence and $p$ is a proposition. In the case where $r = 0$ we have no reliable evidence and are totally ignorant. Now, belief in the truth of a proposition implies an inverse belief in its falsity, and so we can similarly define a *counter-fact*.

$$\lim_{s\to\infty} \text{Belief}(\neg p, s) = \text{Counterfact}(p)$$

where $s$ is the evidence for $\neg p$. Because $\text{Belief}(p, r)$ and $\text{Belief}(\neg p, s)$ are necessarily in an inverse relationship, we see that belief lies on a line between counterfact and fact, which will we call the *polarity* dimension of belief. We can therefore combine these into a single function $\text{Belief}(p, r, s)$.

$$\lim_{(r,s)\to(\infty,0)} \text{Belief}(p, r, s) = \text{Fact}(p)$$

$$\lim_{(r,s)\to(0,\infty)} \text{Belief}(p, r, s) = \text{Counterfact}(p)$$

[11] We are simplifying Kratzer's definition considerably, as she makes a specific theoretical distinction between justified beliefs of *de dicto* (about what is said) and *de re* (about the thing, denoting something in the world) true propositions, only the latter of which are facts. However, this distinction is excessively technical and superfluous for our case.

[12] It may occur to some readers that *theory* is perhaps a more apt term than *belief*. We justify this vocabulary in two ways. First, it is consistent with the language used in [38] and other works in epistemology, that define *knowledge* as justified true belief. Second, *belief* implies a believing agent, while *theory* implies a belief shared by multiple agents, and this distinction will be useful later in Section 3.4 when we introduce a practical application of belief-based factuality to knowledge representation.

While belief in a proposition and its negation are in an inverse relationship, this is not the case for their evidence parameters $r$ and $s$, as it is possible to have considerable evidence both for and against a proposition. Where $r = s = 0$ we have no evidence for either and the truth of the proposition is completely unknown. Where $r = s > 0$ we have conflicting evidence and the proposition is equally likely to be true and not true. This distinction is best illustrated with a frequentist process such as a coin flip. Before any flips are observed ($r = s = 0$) we have no knowledge of whether the coin is biased in any way. If after 1000 flips we observe 500 heads and 500 tails ($r = s = 500$), it is known with high certainty that the coin is not biased.

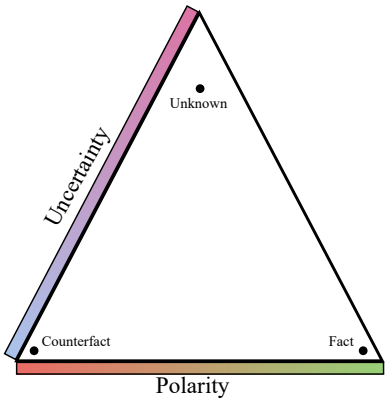



Figure 1.2. Barycentric triangle representation of factuality, with axes for polarity and uncertainty.

Thus the values of $r$ and $s$ comprise an overall uncertainty term, adding an additional dimension to our belief representation. The uncertainty dimension interacts with the polarity dimension such that as uncertainty increases, polarity tends towards a middle value. We can therefore define Belief : $(p, r, s) \rightarrow \triangle^2$ and plot it using barycentric coordinates as in Figure 1.2. We thus arrive at the following definition of *factuality*.

**Definition 1 (*Factuality*)**

*The coordinates* Belief$(p, r, s) \in \triangle^2$ *representing the polarity and uncertainty of a belief in a proposition* p *given the supporting evidence* r *and contradicting evidence* s.

By splitting factuality into the dimensions of polarity and uncertainty our definition matches the natural language used to express it: polarity is expressed using negation and uncertainty is expressed using speculation. Crucial to our argument for the inclusion of these dimensions as part of the core event representation, the absence of any negation or speculative cue implies a positive and certain statement, and so *each and every natural language proposition can be mapped to a factuality coordinate*.

We note that this definition of factuality accords with and is a generalization of previous works on factuality in NLP, which assign values to regions of the $\triangle^2$ simplex. For example, [39] define 6 regions corresponding to Counterfact ($r \ll s$), Doubtful ($r < s$), Possible ($r \approx s$), Probable ($r > s$), Fact ($r \gg s$), and Uncommitted ($r \approx s \approx 0$). It has also been

realized in Subjective Logic Theory [40], which gives a precise definition of the function Belief : $(p, r, s) \rightarrow \triangle^2$. This theory will be summarized in more detail later in Section 3.4.1. Our definition does, however, differ from previous work which treats belief as disconnected from truth [41] as it makes reliable evidence a core aspect of belief. Of course, what is or isn't reliable evidence is highly debatable, but a treatment of this highly philosophical question is out of scope for this work, so we here assume that the reliability of evidence is given without any loss of generality.

## 1.2 Motivation for the Focus on Biomedical Literature

The discussion above is applicable to natural language expressions in general, so it is important to justify this thesis' focus on the biomedical literature. As previously mentioned, we have a special interest in scientific expressions since we feel these stand to benefit considerably from our proposed emphasis on factuality. While scientific NLP is a growing sub-field with a number of datasets (e.g., SciERC [32]), workshops (e.g., SciNLP [42]), and shared tasks (e.g., CL-SciSumm [43]), the overwhelming majority of research focuses on biomedicine. This is perhaps due to the vastness, specificity, and practical clinical relevance of the knowledge contained in biomedical literature which motivates a number of downstream applications. For example, knowledge bases and ontologies such as the UMLS [44], SNOMED CT [45], Entrez Gene [46], and iDISK [47]) can form the backbone of clinical decision support systems which facilitate clinicians' access to evidence-based recommendations at the point of care. However, keeping knowledge bases current is difficult and resource intensive [48]. A system for automatically extracting and collating evidence that is faithful to the inherently uncertain nature of scientific knowledge could ease the maintenance costs of such systems while also expanding their scope. Another application lies in emerging AI-based tools for biomedical question answering and search such as SPIKE[13] and FutureHouse[14], which –like the clinical decision support systems– need to be able to process vast amount of scientific data while respecting contradictory or uncertain knowledge claims. Likely for these same reasons of vastness and practicality, NLP research on negation, speculation, and factuality has historically been concentrated in the biomedical domain, and we will give an overview of this research in Chapter 2.

Ultimately, the motivation for focusing on the biomedical domain is a practical one: the rich history of research on negation, speculation, and factuality in biomedical NLP provides us with a substantial foundation upon which this work aims to build.

## 1.3 Research Aims and Thesis Structure

The overarching aim of this thesis is the development of methods for EE that treat polarity and uncertainty as central aspects of the event representation. Given this, we would like

[13] `https://spike.apps.allenai.org/`
[14] `https://www.futurehouse.org/about`

our models to perform joint prediction of an event's core representation and its factuality dimensions, which we collectively refer to as Factualized Event Extraction (FEE). More specifically, our research aims can be split into two:

- **RA1:** To theoretically and practically demonstrate the importance of treating polarity and uncertainty as central dimensions of the event representation.
- **RA2:** To develop novel machine learning (ML) models for FEE that not only achieve good prediction performance but also learn internal representations of factuality that align with Definition 1.

We have already provided one argument for RA1 in Section 1.1. The remained of this thesis continues to address these aims with the following structure. Chapter 2 provides a literature review of the history and current state of EE and context classification (CC) in biomedical NLP –the latter review focusing on negation, speculation, and factuality. We also include a summary of relevant datasets and a discussion of their limitations. Building off this discussion, Chapter 3 introduces SemMed Context, a new large dataset of biomedical events annotated for polarity and uncertainty using high-accuracy automated methods. This chapter includes the evaluation of a baseline model for FEE on SemMed Context (RA2), as well as a proof-of-concept knowledge graph demonstrating the utility of representing events as beliefs (RA1). Extrapolating from Section 1.1 above, Chapter 4 theorizes that polarity, uncertainty, and core propositional content can be modeled as entirely independent variables, testing this theory by estimating a Variational AutoEncoder (VAE) on both general-domain text and SemMed Context (RA1 and RA2). Following this, Chapter 5 switches to a more practical focus, aiming to achieve high prediction performance on SemMed Context. It introduces Levitated Context Markers (LCMs), a novel model for FEE (RA2). Chapter 6 finishes by summarizing our findings, discussing the overall limitations of this thesis, as well as what our findings suggest regarding future research in this area. Additionally, we include two appendices that describe relevant novel works completed as part of this thesis, but which are not central to it. Appendix A introduces a novel method for representing and learning from disagreement among annotators, and Appendix B describes two Python packages for streamlining machine learning research.

# Chapter 2

# Background

This chapter provides a high-level overview of the Event Extraction (EE) task. A brief preliminary first defines Event and Context in terms of the standard linguistic concepts of predicate and adjunct expressions. The following section provides the definition of EE that will persist throughout this thesis, and describes its relationship to NLP tasks such as relation extraction (RE). It then describes the domain-specific task of biomedical EE, and provides a chronology of methods from early rule-based systems up to current methods based on pre-trained language models (PLMs). The chapter then moves to a definition of the event context classification (CC) task, including a survey of previous work on negation, speculation, and factuality classification, and closes with a brief overview of existing datasets for CC. This chapter is meant to provide a high-level overview of the research in this area, while later chapters include a more detailed summary of previous work as it relates to the methods under consideration.

## 2.1 Linguistic Preliminaries

As discussed in the introduction, the goal of Contextualized EE is to extract structured representations of propositions including any contextual information. To better understand the desiderata for these representations and therefore the precise targets of the event extraction and context classification tasks, we here build up to formal definitions of Event and Context Dimension via the core concepts of predicate, argument, adjunct, and proposition.

A **predicate** is an expression which describes an action. It may be a single word (e.g., "I *ate* toast.") or a phrase (e.g., "The film *is made even more enjoyable* by the great acting"). All predicates take one or more **arguments**, which are expressions that are necessary to complete the meaning of a predicate, such as the subject and object. A predicate with all of its arguments is a **complete predicate**. We define an event in terms of a complete predicate.

**Definition 2 (*Event*)**

*A complete predicate expressed in text which describes a world state.*

According to this definition, an "Event" is simply something that happens in a real, imaginary, or hypothetical world. For example, "*Colonel Mustard killed Professor Plum*" is an event within the imaginary world of Cluedo. Now, the meaning of a complete predicate (and

therefore an event) can be modified by one or more optional adjunct expressions, and a complete predicate with zero or more adjuncts is a **proposition**. To extend the Cluedo example, "*Colonel Mustard killed Professor Plum with the candlestick in the library.*" is a proposition that contains two adjuncts to the complete predicate. These adjuncts express instrumental and locative categories of meaning, but many others are possible (cf. [49] for a more detailed discussion of categories of adjuncts). Each adjunct category is a context dimension.

**Definition 3 (*Context Dimension*)**

*A category of adjunct expressed in text that modifies an Event.*

Thus by "Context" we mean the collection of adjuncts attached to an event. Again, many context dimensions are possible and have been studied in previous work, but as motivated in Section 1.1 this thesis focuses on the context dimensions of negation and speculation as fundamental determinants of the expressed factuality of events.

## 2.2 Event Extraction

Given Definition 4 above, Event Extraction is the task of identifying and representing world state descriptions in text. This preliminary definition raises the question of how events should be represented. As events are predicates, it is natural to develop an event representation schema that accords with the structure of the predicate arguments. An EE system, therefore, must take text as input, identify entities of interest, and determine what, if any, relation exists among them according to the given event schema (cf. Figure 2.1). A variety of schemas are possible, depending on the goals of the system and the argument structure of the target predicates, but one of the most ubiquitous is the *subject-verb-object* triple (e.g., *Colonel Mustard-kill-Professor Plum* from "*Colonel Mustard killed Professor Plum*"). This representation is so common, in fact, that the task of identifying the relationship between two entities is treated as its own task, called *Relation Extraction* (RE). For example, the ACE 2005 shared tasks treat RE and EE as distinct [10]. This distinction has been historically useful: while RE and EE both deal with world-state descriptions, events can be arbitrarily complex and thus much less restrictive than the *subject-verb-object* relation triples, meaning that methods can vary widely between the two.

However, because this thesis is concerned with adding context to events generally, it is necessary to consider them from the most simple –comprised of a single text span– to the most complex –including nested events. Thus while a distinction between EE and RE has been useful in the previous literature, this thesis treats RE as simply one instance of EE. Our definition of EE is, therefore, suitably generic to cover the multitude of event representations found in the research literature (Definition 4).

**Definition 4 (*Event Extraction*)**

*The task of constructing structured representations of world state descriptions in text by matching them to a predefined schema.*

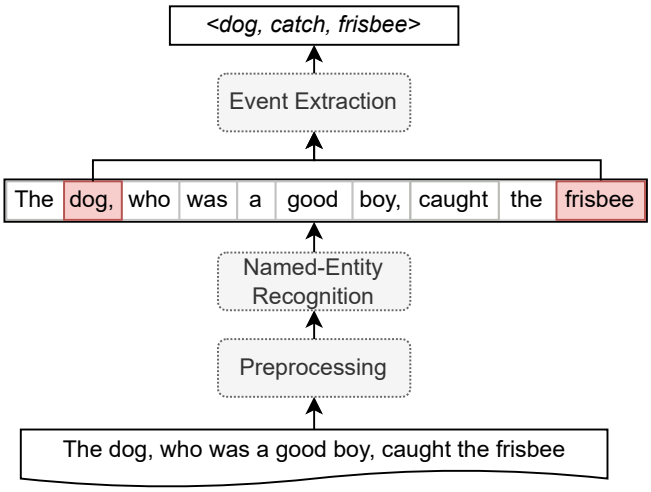



Figure 2.1. A high-level overview of an EE system. The input text is preprocessed (tokenization, normalization) and sent to a named-entity recognition (NER) module which identifies all the entities. The EE module predicts the event structure according to the representation schema. In this example, which uses a <*subject, verb, object*> schema, the EE module would predict which of each pair of entities is the *subject* and *object*, respectively, as well as the *verb* that connects them.

Biomedical EE simply constrains Definition 4 to the clinical, biological, and health science domains, with the aim to improve the accessibility of knowledge contained in clinical notes and scientific reports. While the methods used for biomedical EE are largely the same as those for the general domain, biomedical text introduces a number of new difficulties. First, the language of biomedicine has been shown to vary considerably from general language [50], meaning that methods developed for general-domain EE can transfer poorly. Second, understanding biomedical text requires a significant amount of background knowledge, and what might be considered "common sense" in the biomedical domain might be considered obscure to a non-expert. Third, the language and background knowledge required are not uniform across sub-domains within biomedicine: what is required to understand an *in-vitro* study of cancer cell growth is likely very different from that required to understand the results of a clinical trial of a new cancer treatment, which is again very different from understanding a care report of a cancer patient. In fact, it is challenging even to apply an NLP system developed at one clinical site to another [51]. While biomedical EE is often motivated by progress in the general domain, these difficulties mean that it is problematic to generalize the performance of one EE system across domains. The following sections therefore describe both general- and biomedical-domain methods for EE, highlighting the current state-of-the-art (SOTA) biomedical systems.

### 2.2.1 Methods for Event Extraction

The history of EE methods can be broadly split into three major periods: rule-based and classical machine learning (ML) methods, deep neural networks (DNNs), and pre-trained language models (PLMs). Across these periods there are a number of milestones in EE model development as research took advantage of advancements in NLP and ML. These milestones include multi-task learning (MTL) in 2011, structured prediction in 2013, recurrent neural networks (RNNs), convolutional neural networks (CNNs), and the attention mechanism in

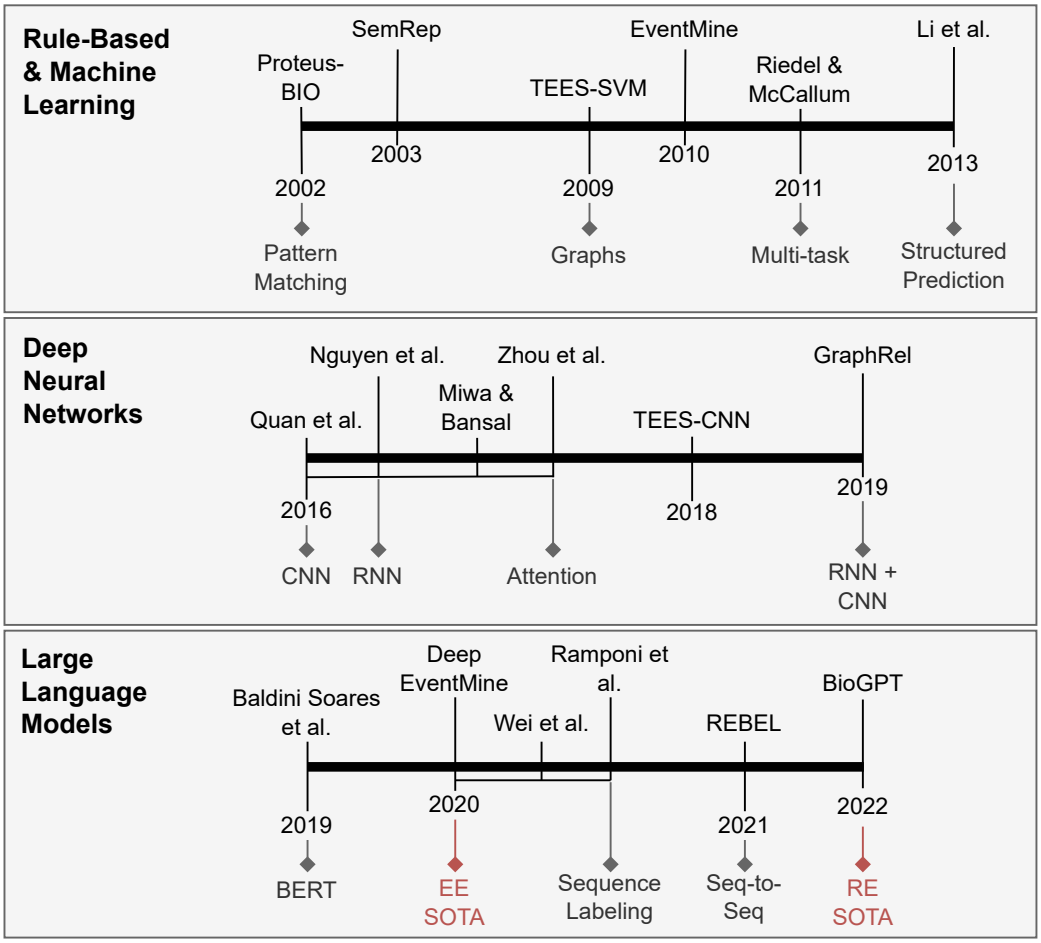



Figure 2.2. A timeline of research milestones in event extraction, split up into the three major periods: rule-based/machine learning, deep neural networks, and large pre-trained language models. The current state-of-the-art methods for relation extraction (RE) and event extraction (EE) are indicated in red.

2016, BERT in 2019, and sequence-to-sequence (seq-to-seq) transformer decoder models in 2021. A timeline of advancements is given in Figure 2.2.

This timeline makes it clear how modern models evolved. The rigid, hand-defined rule-based systems in the early 2000's led to more flexible classical ML models with hand-defined features. Early structured prediction models (such as conditional random fields) which used these hand-engineered features were a natural precursor to RNNs, which relaxed the Markov assumption and reduced the need for feature engineering. The attention mechanism –first introduced as a component in RNN models– became the core component of the Transformer, which formed the foundation of PLMs such as BERT and GPT [52], which are the backbone of the current SOTA in EE. The following sections describe each of the three periods of EE research in greater detail. EE systems are built and thus dependent upon accurate identification of entity mentions in the text. Named-entity recognition (NER) is its own subfield of NLP with its own methods and challenges. Therefore, when discussing evaluation results below we report the metrics computed using gold-standard entity annotations wherever possible. Where this is not possible, we use "end-to-end" to refer to systems that perform both NER and EE.

#### Rule-Based Systems

The earliest systems for biomedical EE were rule-based. One of the first, Proteus-BIO, maps text to events via a set of manually defined patterns such as `<disease>` killed `<victim>`, utilizing text preprocessed into verb phrases [53]. Another early system, SemRep, uses a rule-based approach to detect relations between entities [54], [55]. Hand written rules map the part-of-speech tagged and dependency-parsed input to a set of predicate types in the UMLS ontology [44]. However, due to the inherently complex and varied nature of how events are expressed in language, these early rule-based systems obtained sub-par performance: Proteus-BIO obtained 0.79 precision but only 0.41 recall on a small test set of only 32 documents. Evaluated on a larger set of 1,371 events, SemRep obtained only 0.55 precision and 0.34 recall when evaluated end-to-end [56]. Nevertheless, rule-based approaches continued to be developed due to their speed and interpretability. More recent rule-based systems generally follow the framework of pattern matching on the preprocessed input text. [57] build on to an existing rule-based semantic parser, collecting data from multiple high-quality ontologies available for the biomedical domain to define schemas for the events in question. [58] and [59] automatically generate an entity dictionary and event patterns from annotated events using a set of hand-defined features. Unfortunately, these systems, like their predecessors, also suffer from low recall. For example, [57] achieve 0.72 precision but only 0.13 recall on the BioNLP'09 dataset, and [59] obtain 0.61 precision but only 0.41 recall on BioNLP'13, both evaluated end-to-end.

#### Machine Learning Methods

Concurrent with the rule-based approaches described above were developments using standard ML models with hand-engineered features. While these features shared much with the hand-defined patterns in the rule-based models –generally focusing on lexical and syntactic attributes of the input– using ML to automatically learn weighted combinations of the features made them more flexible than the rule-base systems. One of the earliest ML models for EE as well as the best performing system in the first BioNLP shared task [60] was the TEES-SVM system, which treated EE as a graph construction problem in which nodes are event trigger spans and edges are arguments [61]. They used multi-class support vector machines (SVMs) for both identifying trigger spans in text as well as predicting edges between each pair of triggers using a variety of lexico-syntactic features. The edge detection step did, however, allow improperly formatted events, so a rule-based post-processing step was used to combine and prune edges according to the annotation specifications as well as remove events without any arguments. A similar SVM-based system was developed in [62], which improved on previous work by optimizing the SVM parameters and performing automatic feature selection. Also building on the work of [61], [63] developed EventMine, which uses an additional SVM in place of the rule-based component to predict complex event structure. This model is trained using positive and negative examples automatically extracted from the training data. EventMine was further enhanced in [64] by adding a rule-based coreference resolution

module and a domain adaptation modules, which improved the performance further.

All of the examples above utilize a hybrid machine-learning/rule-based approach to some degree. In fact, the best performing model in the BioNLP 2011 shared task utilized a hybrid ML and rule-based approach to exploit multiple views on the data [65]. This approached combined the output of multiple existing systems by optimizing an objective function utilizing features computed from the predictions of each model.

A problem with these pipeline systems, however, is that errors made early in the pipeline cascade through to the end. To combat this, [66] developed a model which predicts event triggers and arguments jointly, optionally enforcing constraints on the event structure at the cost of model complexity. While even the simplest of these models, without any constraints on event structure, achieved competitive performance on BioNLP 2009 and 2011, the model which enforced the most constraints on the output was the SOTA at the time of publication.

A major problem with the systems above is their reliance on local classifiers to make predictions. That is, the ML models used in these systems make each trigger and argument prediction entirely independent of the predictions that came before it. Natural language, however, contains dependencies that must be accounted for to make accurate predictions. To address this gap, [67] was the first to treat EE as a structured prediction problem, using a structured perceptron [68] to jointly predict event triggers and arguments. In a structured perceptron, the current prediction depends not only on the input features, but also on the previous prediction. This results in a much larger search space for optimization than with local classifiers, and the authors utilize beam search to obtain an inexact solution to the optimization problem. The authors improved this model further in [69] with new features and a more efficient incremental decoding strategy, achieving SOTA on the general-domain ACE EE dataset [10] at the time of publication.

#### Deep Neural Networks

The importance of treating EE as a structured prediction problem lent itself well to developments in DNNs in the 2010s. DNNs improved upon standard ML approaches in two main ways: First, DNNs are universal function approximators, automatically learning a non-linear projection of the inputs into a new multi-dimensional space that is a good predictor of the classification target [70]. This is in contrast to the generally linear classical ML methods[1]. Second, DNNs automatically learn this non-linear projection via gradient back-propagation of the error function, meaning that, unlike classical ML methods, they do not require hand-engineered features and can instead learn from incredibly simple data encodings such as binary word vectors.

Together these improvements led to the development of word embeddings, continuous vector representations of words learned from raw text. Word embedding methods such as word2vec [71] and GloVe [72] work by defining a simple prediction task –for example, word2vec

[1]Non-linear decision boundaries are of course possible using kernel functions in SVMs, but these are still limited to a restricted set of transformations and require hand-tuning to achieve good results.

predicts a target word from its context words– and training a DNN using a large amount of text data. When training is complete, each word in the dataset is paired with its learned continuous representation to form a lookup table. Despite the simplicity of these models, word embeddings have been shown to encode a variety of lexical and semantic similarities between words [71], [73], and initializing DNN models with word embeddings in place of binary vectors results in impressive performance gains [74][2].

Taking advantage of these improvements, the seminal research in [76] showed that Convolutional Neural Networks (CNNs) –a DNN architecture originally developed for image processing– were also powerful models of language. CNNs work by first learning local features in a small window around each word in the input using a convolutional layer and then successively combining these local features into higher level features via a stack of convolutions. [77] were the first to apply CNNs –and DNNs in general– to biomedical EE, achieving a 3-5 percentage point increase over the previous SOTA SVM on two datasets. Their model employed a CNN on top of word embeddings learned from PubMed, PMC, MEDLINE, and Wikipedia. An extension of the SVM-based TEES-SVM model, TEES-CNN predicts event triggers and arguments in a pipeline using word embeddings as well as lexico-syntactic features as model inputs, achieving SOTA on 7 biomedical EE datasets [78]. [79] also use dependency parses as input to a CNN, but rather than extract features, they use the order of words in the dependency graph to learn the word embeddings.

Alongside CNNs, recurrent neural networks (RNNs) were also making strides. As time-series models, RNNs are an obvious choice for modeling natural language. They also remove the Markov assumption required by classical ML models such as Hidden Markov Models and structured perceptrons, meaning that they can learn from longer histories. The Long Short-Term Memory (LSTM) variant of the RNN has been especially successful in NLP. One of the first to apply RNNs to EE was [80], who utilize an LSTM and binary memory vectors to jointly predict triggers and arguments. Concurrently, [81] developed an model that outperforms the SVM model of [69] by 3-5 percentage points depending on the class. Theirs was a hierarchical model in which one LSTM detects event triggers which are then passed to a second LSTM which predicts the event structure.

The LSTM model developed by [82] was the first for EE to utilize an attention mechanism. The attention mechanism, generally, is a method for determining a soft alignment between sequences and was used in this case for automatically determining which words in the input are the most important for classification without any additional external knowledge or error terms. More formally, attention is a learned weighted average over the members of a sequence. In additive attention, introduced in [83], a neural network $f(\cdot, \cdot)$ computes alignment scores $\boldsymbol{E} \in \mathbb{R}^{L \times M}$ between two sequences $\boldsymbol{a} = [a_1, ..., a_L]$ and $\boldsymbol{b} = [b_1, ..., b_M]$ (Equation (2.1)). These scores are then normalized into attention weights $\boldsymbol{\alpha}$ using softmax (Equation (2.2)) and used to compute a representation for each token $h_i$ as a weighted average of the input (Equation (2.3)).

[2]Although it should be noted that follow-up work attributed the performance gains of word embeddings to other hyperparameters and design choices [75].

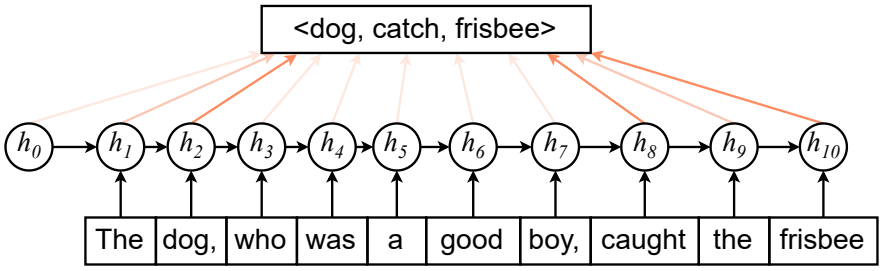



Figure 2.3. A simplified example of how an attention mechanism works for relation extraction. Given the sentence and a target relation triple, the attention mechanism learns weights for each token such that when added together they are a good predictor of the target. Higher attention weights are given darker orange highlights.

$$e_{ij} = f(a_i, b_j) \tag{2.1}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j=1}^{L} \exp(e_{ij})} \tag{2.2}$$

$$h_i = \sum_{j=1}^{L} \alpha_{ij} a_i \tag{2.3}$$

In theory, relevant data points are given high attention weights, while irrelevant data points are zeroed out. An example of the attention mechanism for relation extraction is given in Figure 2.3.

[84] also developed an LSTM-based model for biomedical EE, which utilizes Abstract Meaning Representations (AMR) [85]. AMR is a framework for constructing formal representations of the semantics of natural language sentences that abstracts away from lexical and syntactic variation, encoded as directed acyclic graphs. This paper made two major contributions: 1) they show that events are subsets of the AMR sentence representation; 2) they cast EE as a subgraph identification problem in which an LSTM is trained to determine the subgraph of an AMR that corresponds to an event. While this model failed to outperform the SOTA, it did point the way forward for neural graph-based methods in EE.

Combining LSTMs and CNNs, GraphRel stacks an LSTM and a Graph Convolutional Network (GCN) to jointly predict entities and relations [86]. GCNs are a generalization of CNNs which perform convolutions over neighboring nodes [87]. GraphRel takes as input a sentence and its dependency parse. The LSTM and GCN each generate word features from the input sentence and dependency graph, respectively, which are then combined and passed to a second GCN that makes a relation prediction for each word pair. This model achieved SOTA on two general domain datasets. [88] also developed a graph-inspired LSTM model for RE. Their model makes use of the attention mechanism from [82], while adding a relation aggregation layer that infers relations between entities by walks along existing edges between them. This model was extended from sentences to full documents in [89] and outperformed the SOTA. It was also applied to the biomedical domain for predicting relations between

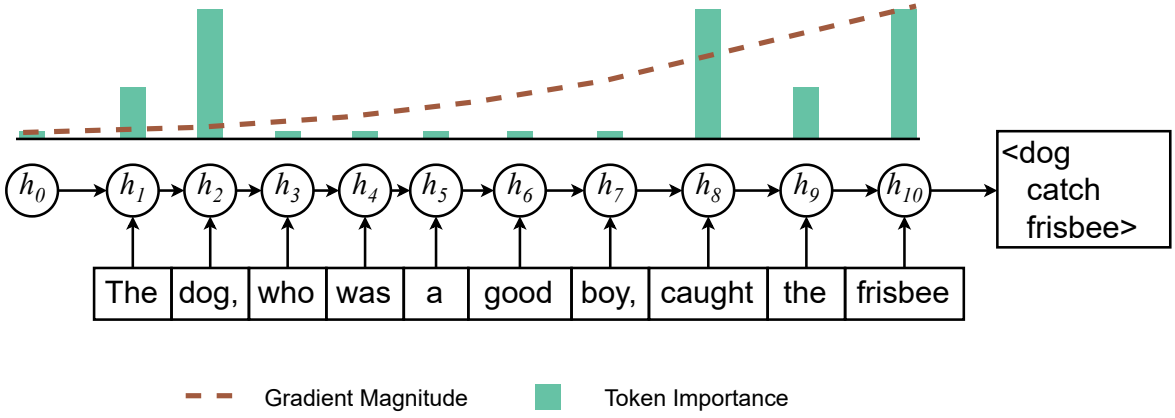



Figure 2.4. A depiction of the vanishing gradient issue in RNNs. The sentence representation is computed as the last hidden state $h_{10}$ of the input sequence. Given the vanishing gradient, one of the most important input words *dog* has little effect on the final representation.

drugs in clinical text, coming in first place in the 2018 n2c2 shared task[3] [90].

RNNs, despite their impressive performance, have three major flaws. The first is the **vanishing gradient** problem. Because the representation of the token at the current time step, denoted $x_t$, is computed recursively from all the previous token representations $x_{<t}$, the gradient computed during model training becomes smaller and smaller as it goes back in time. In other words, the farther away a given token is from the target $x_t$, the less influence it has. This effectively forces the model to compute the representation for $x_t$ from the closest tokens, which can be problematic for long or complex sentences. An example of this issue is illustrated in Figure 2.4. The vanishing gradient problem is related to the second flaw: **memory compression**. RNNs recursively combine the input into a single representation, which can be too small to accurately encode the sequence, especially in combination with the loss of information in early time steps due to the vanishing gradient. The third flaw is simply that RNNs are **directional**, meaning that they only compute the representation of $x_t$ from $x_{<t}$, even though words in natural language often have dependencies both before and after. For example, in *"I went to the bank to untie the boat."* the meaning of the homograph *bank* is determined by the words that come after it, rather than before.

These issues do have solutions: vanishing gradients and memory compression were less of an issue with LSTMs and their forget gates, and directionality could be addressed with bidirectional models (albeit at the cost of larger model size). While CNNs do not necessarily suffer from these issues, they do lack the position awareness of RNNs. That is, CNNs learn to recognise a certain pattern in the input text, no matter which words comes before or after it. But it turns out that the attention mechanism, a component of previously SOTA RNN models ([82], [89]) had the potential to address all of these issues.

#### Pretrained Language Models

As its name suggests, the seminal paper "Attention Is All You Need" showed how to entirely replace RNNs and CNNs with the attention mechanism [91]. They propose an encoder-

[3] https://n2c2.dbmi.hms.harvard.edu/2018-challenge

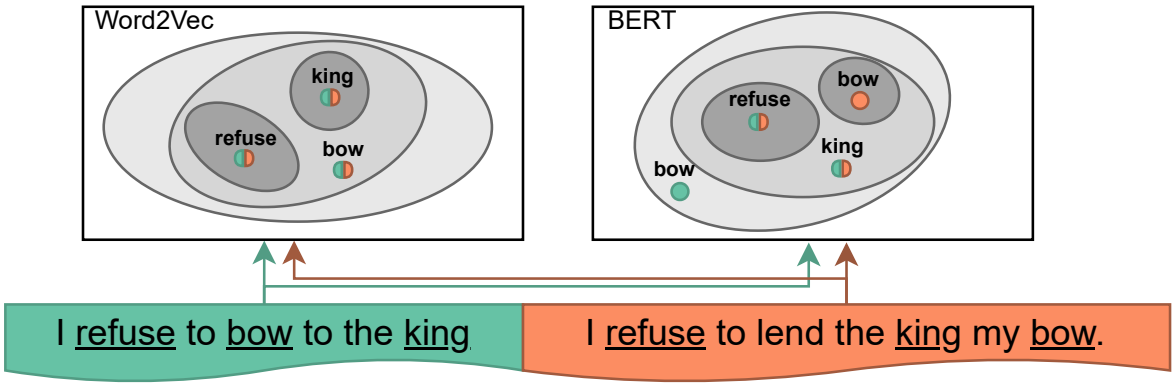



Figure 2.5. A comparison of the token embedding spaces of non-contextual models such as Word2Vec to contextual models such as BERT. Word2Vec is a simple key-value table of tokens to embeddings, meaning that homographs such as *bow* are encoded the same despite their very different meanings. BERT, on the other hand, computes token embeddings dynamically according to the other tokens in the input. This means that homographs such as *bow* can be encoded differently. Note that *refuse* is also a homograph, but only one of its meanings is used here so it is always encoded the same.

decoder model called a *Transformer* that is simply a stack of layers each of which computes attention between the input and itself. This is called self-attention and was first introduced as a layer in an LSTM model [92]. The encoder performs self-attention on the input to produce a hidden representation $\boldsymbol{z}$ of each token, while the decoder performs cross-attention with $\boldsymbol{z}$ and self-attention over the input, which is shifted and masked so that the output can only be conditioned on the tokens that came before it. In comparison to RNNs, self-attention in the transformer encoder is inherently non-directional, and because it dispenses with recurrence altogether it does not suffer from the vanishing gradient or memory compression problems.

As part of their research, the authors of [91] introduce scaled dot-product self-attention, which is more efficient than the additive attention as given in Equations (2.1) to (2.3) thanks to fast matrix multiplications. Given an encoded input sequence $\boldsymbol{a} = [a_1, ..., a_L]$ with each member $a_i$ of dimensionality $d_e$, scaled dot-product attention is computed as

$$Q = \boldsymbol{a}W^Q, \quad K = \boldsymbol{a}W^K, \quad V = \boldsymbol{a}W^V \tag{2.4}$$

$$E = \frac{QK^T}{\sqrt{d_k}} \tag{2.5}$$

$$A = \text{softmax}(E) \tag{2.6}$$

$$H = AV \tag{2.7}$$

where $W^Q \in \mathbb{R}^{d_e \times d_k}$, $W^K \in \mathbb{R}^{d_e \times d_k}$, $W^V \in \mathbb{R}^{d_e \times d_v}$. First, the inputs are linearly transformed into (Q)uery, (K)ey, and (V)alue matrices. The dot product between $Q$ and $K$ gives the similarity scores $E$, which are scaled by $d_k$ to improve gradient stability during training. $E$ is then normalized via softmax to obtain the attention weights $A \in \mathbb{R}^{L \times L}$ between each pair of tokens in the input. Finally, these are transformed into $H$ which is the same dimensionality as the inputs. Scaled dot-product attention can also be easily parallelized by a simple linear projection of $Q$, $K$, and $V$ into $h$ different *heads*. These multi-headed attention

layers, stacked on top of each other, comprise the encoder and decoder of the transformer model.

The output $H$ of the scaled dot-product self attention is the same length as the input. In other words, it is essentially an input embedding. Therefore, by training a transformer model on a task similar to those used by word embedding models, the resulting model can then be used to generate word embeddings. This was the advent of the pretrained language model (PLM), which forms the backbone of many of the SOTA NLP systems.

One of the most successful PLMs is BERT (Bidirectional Encoder Representations from Transformers) [93], which is precisely the model introduced in [91] pretrained on a large amount of general-domain text data. Where word embeddings such as word2vec or GloVe are essentially lookup tables, BERT can be viewed as a dynamic contextual encoder, outputing different embeddings for the same token depending on the surrounding words. This allows BERT to properly account for polysemous or homographic words where previous word embedding methods could not. This is illustrated in Figure 2.5. BERT is also adaptable and easily extensible: its output can be used directly for prediction or additional layers can be stacked on top for more complex tasks. Soon BERT variants were developed that were pretrained on different data such as scientific (SciBERT) [94], biomedical (BioBERT [95], PubMedBERT [96], BlueBERT [97]), and clinical (ClinicalBERT [98]).

These pretrained models, like the word embeddings that they supplanted, kick-started model training by implicitly modeling basic linguistic knowledge [99], [100]. As the original BERT paper showed, appending one or more task specific layers to these models and running a relatively short bout of training (a process called fine-tuning), it was possible to obtain a high-performing model quickly. For example, fine-tuned BERT models achieved new SOTA results on the 9 datasets in the General Language Understanding Evaluation (GLUE) benchmark, beating out previous SOTA models by multiple percentage points.

Given its successes, it was only natural to apply BERT to EE as well. The first exploration of BERT for EE was [101], who experimented with different task representations for learning relations between entities. The best performing method according to their experiments involved marking up the input text with tokens to specify the subject and object spans, then using the learned representations of these markup tokens to predict the relation type. This model outperformed the previous SOTA by 2-17% across four general-domain datasets, illustrating the large gains to be had by using PLMs for EE.

[102] introduce a variation on the markup tokens used by [101] called Levitated Markers. Instead of marking up the input text directly, the marker tokens are appended to the end of the input and the attention mechanism is modified to allow the marker tokens to "see" the text, but not the other way around. Their method allow efficient batching by marking up all the entities in a sentence, vastly improving model speed compared to model that creates separate training examples for each entity pair. This method also outperforms the SOTA, as they hypothesize that batching all entities together allows the model to learn use correlations between them. A follow up work combined solid, in-text markup tokens with levitated

markers. The subject span is given a solid marker while all candidate object spans are given levitated markers [103]. This subject-oriented markup strategy again allows even more useful correlations to be learned between possible objects further improving performance.

In the biomedical domain, [104] were the first to apply PLMs to EE. They used a variation on the solid markers of [101] that also replaced the entity mentions in the input text with their type (e.g., `@Drug$`). This model outperformed the previous state of the art on two clinical RE datasets. Reframing EE as a sequence labeling –rather than a structured prediction– task, [105] attempted to take advantage of BERT's power to simplify the task: they flatten complex event structures into three sets of sequence labels representing triggers, arguments, and relations. A multi-task learning (MTL) layer on top of BERT jointly predicts the sequence labels for each input token. Their experiments show a 1.5% F1 score increase over the previous SOTA on the GENIA dataset, with improvements of up to 13% on certain event types. DeepEventMine [106] is a refactoring of EventMine [63], [64] that is based on SciBERT. The SciBERT-encoded input is passed to three subtask layers: entity/trigger, role, and event. The entity/trigger layer finds all possible entity mentions in the input, including nested events. The role layer assigns a role to each entity pair, utilizing output from the SciBERT layer as additional context. Finally, the event layer constructs events from all possible combinations of role pairs. Overall, DeepEventMine outperformed the previous SOTA on 7 biomedical EE datasets and is still holds the current SOTA at the time of writing.

The encoder-decoder architecture in the transformer model described above lent itself well to paired sequence classification tasks such as natural language inference and machine translation where each sequence was relatively short. Longer sequences, however, are prohibitive due to the computational cost of modeling the entire input sequence at once. The transformer decoder [107] addresses this by making two simple changes to the original transformer architecture: they drop the encoder altogether and concatenate the input sequence pair into a single sequence separated by a special token. The transformer decoder is thus an auto-regressive language model, simply predicting the next word from the words that come before it and implicitly learning to perform NLP tasks via a large amount of pre-training. The most successful implementation of this has been the GPT (Generative Pretrained Transformer) series of models, which are auto-regressive transformers trained on web-scale data and achieve impressive zero-shot performance on a number of NLP tasks [52], [108] . However, because GPT is a language model designed to output text, it is not ideal for classification or structured prediction tasks.

Models such as T5 [109] and BART [110] re-couple the transformer encoder and auto-regressive decoder into models ideal for sequence-to-sequence transduction tasks. The encoder enables them to take advantage of both left and right context (unlike GPT), while the auto-regressive decoder is ideal for generating text. They are designed as *denoising* encoder-decoder models, meaning that they are tasked to reconstruct the input that has been corrupted in some way. T5's denoising pre-training objective is to predict token spans that have been randomly masked in the input, while BART's objective combines this with the reordering the randomly shuffled input sentences. The authors of both the T5 and BART papers show

impressive performance on the GLUE benchmark, with BART setting a new SOTA.

The successes of these models motivated another new approach to EE in which the model is trained to output natural language describing the event, rather than a structured prediction. One of the first explorations of this was [111], who fine-tuned BART on 7 different RE datasets. Their REBEL system works by first linearizing the entity and relation representations to form a suitable target for the seq-to-seq task and then simply training the model to predict the linear representation. An example of linearization is given in Figure 2.6. Most recently, BioGPT [112] acheived SOTA performance on three biomedical RE datasets using a GPT model pre-trained on a large amount of biomedical literature. Their linearization approach is based on prompting. Given the <*source*> text and a task-specific <*prompt*>, the <*target*> output is *"The interaction between <entity1> and <entity2> is <relation>.*. During training, the model is tasked with reconstructing the <*source;prompt;target*> input, while during inference it is provided with only <*source;prompt*> in order to generate <*target*>.

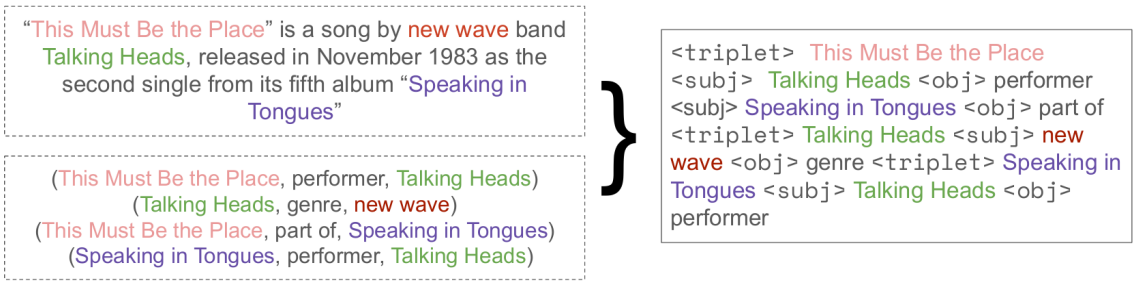



Figure 2.6. Example of relation triple linearization. Reproduced from [111].

### 2.2.2 Takeaways

Overall, biomedical EE research has continued to take advantage of advancements in NLP and ML to improve performance. However, the task is far from solved. The current SOTA performance on the GENIA EE task by DeepEventMine is only 0.64 F1. The results for end-to-end systems on biomedical tasks are even lower, with BioGPT obtaining between 0.38 and 0.45 F1 on three drug interaction tasks. These results highlight just how difficult these tasks are, even for powerful models pre-trained on large corpora.

## 2.3 Context Classification

The EE models discussed in the previous section were all restricted to extracting the events only, without concern for any context expressed in adjunct expressions. However, as discussed in Chapter 1, a complete understanding of an event often requires additional contextual information. The task of Context Classification (CC) is precisely to extract this context.

**Definition 5 (*Context Classification*)**
*The task of predicting the values of one or more context dimensions for a given event from the*

*text.*

This thesis is primarily concerned with the context dimensions of polarity and uncertainty as these are the determinants of an event's expressed factuality. For example, the event representation in Figure 2.9c states that 5-LOX inhibitors influence NF-kappaB induction, but the sentence actually expresses that this is *not* the case. A correct representation including the negation cue is depicted in Figure 2.7. Unfortunately, these dimensions are often overlooked when designing EE tasks and, as mentioned in Chapter 1, many EE datasets simply to do not annotate negated or uncertain event mentions. EE models trained on these datasets thus can not learn to distinguish positive polarity or certain events from their opposites, which can lead to false positive predictions. More importantly, the ability to extract negated and speculative events is crucial for downstream knowledge applications as they may be included in the evidence for or against a claim. For example, a system for determining the overall consensus of a knowledge claim from the scientific literature needs to collate both negative and positive claims regarding a given hypothesis event.

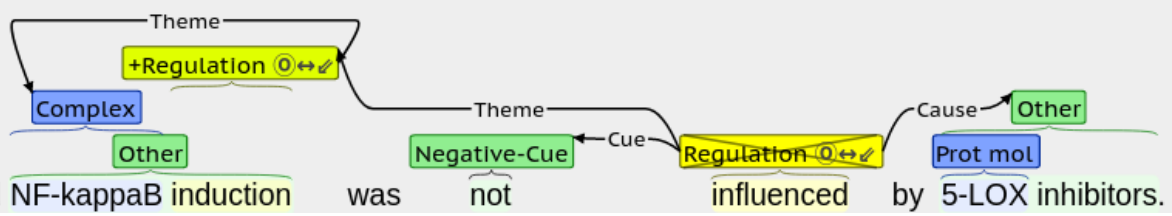



Figure 2.7. The GENIA example from Figure 2.9c with the addition of the negation cue.

This section therefore provides a summary of previous work on negation and speculation processing with an emphasis on systems that link these dimensions to events. This summary also includes work on factuality detection, which –as shown in Section 1.1– can be decomposed into negation and speculation detection.

### 2.3.1 Negation and Speculation Processing

Negation and speculation processing can be broadly divided into two subtasks: scope resolution and event context classification. Scope resolution is the task of determining which words in the input fall under the influence of a given negation/speculation the cue. For example, in *"I ordered the chicken, not the steak."*, only *"the steak"* falls under the scope of *"not"*. Scope resolution therefore relies on the upstream cue detection task, which predicts whether a given token indicates negation or speculation (e.g., *not, never, possibly, maybe*). Scope resolution can be performed in reference to structured events, but it is more often treated as a token prediction task, simply predicting whether each token is scoped under a given cue.

#### Scope Resolution

The majority of the work on negation and speculation processing focuses on scope resolution without consideration for events. One of the earliest and most successful systems for negation

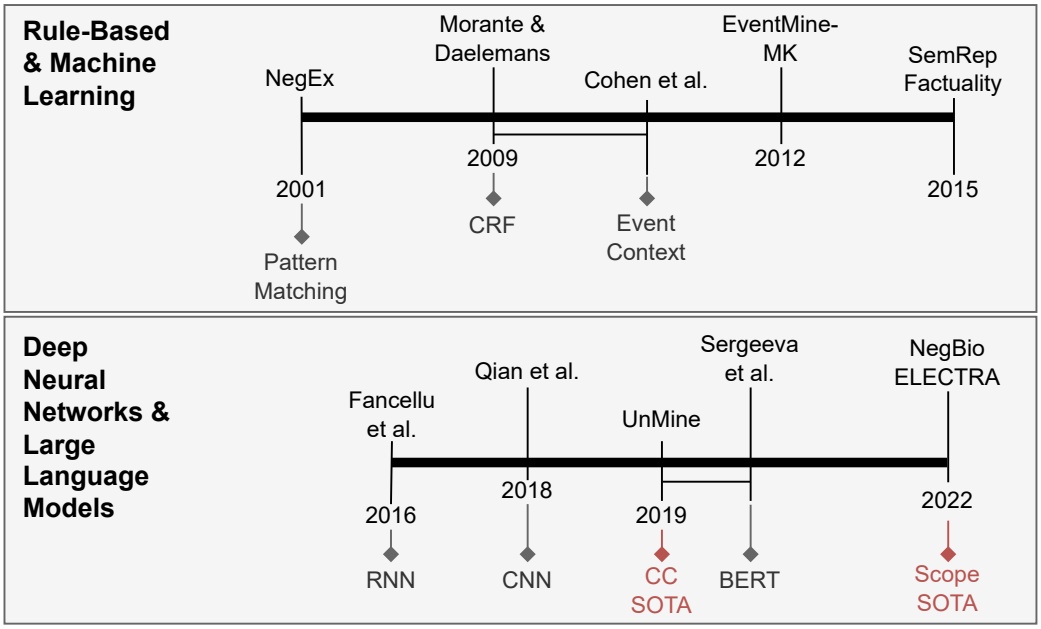



Figure 2.8. A timeline of research milestones in negation and speculation processing, split into periods akin to those in Figure 2.2: rule-based/machine learning, deep neural networks, and large pre-trained language models. The current state-of-the-art methods for scope resolution and event context classification are indicated in red.

processing was NegEx, a regular expression-based system developed for identifying negated phrases in clinical discharge summaries [9]. Despite its simplicity, NegEx achieved a precision of 0.95 in their evaluation, with a sensible recall of 0.77. However, the use of regular expressions limits NegEx in cases where the cue is far from the entity in question. A more modern rule-based system is NegBio [113], which addresses this by defining patterns over dependency parse graphs. It also defines patterns to detect speculation. Similarly, DEEPEN [114] adds dependency information to NegEx, improving precision on two clinical datasets, but with a drop in recall.

ML-based approaches include [115] and [116] who trained conditional random field (CRF) models for identifying negation and speculation cues and scopes in the BioScope corpus. Improving on this early system, [117] also trained CRF models to detect cues and their scopes in BioScope, but utilized a larger portion of the training dataset than [115], outperforming their system by 15% F1. [118] combine ML and rule-based approaches, augmenting patterns over dependency trees with an SVM candidate ranking model to achieve SOTA performance on scope resolution in BioScope. Additional performance improvements on BioScope were achieved by [119], who used an SVM with a tree kernel and parse based features.

Like EE, negation and speculation processing benefited from both RNN and CNN based approaches. The first work to apply DNNs to negation processing was [120]. They used a simple bi-directional LSTM (BiLSTM) to resolve negation scope in the general-domain CD-SCO corpus [121] and outperformed the previous CRF-based SOTA by 6%. [122] were the first to employ a CNN model for negation and speculation scope resolution. They augmented their input with hand-engineered parse-based features for negation and speculation scope resolution on BioScope. However, their model beat SOTA only on the speculation task over paper abstracts, performing worse than previous rule-based and ML approaches

on full papers. They explain that the full papers require the model to process long-range dependencies, for which a CNN with its limited kernel size is unsuitable.

Again, negation and speculation processing benefited from the head-start offered by PLMs. The first application of PLMs to negation and speculation processing was [123] who fine-tuned BERT on both biomedical and general domain datasets, achieving a new SOTA over the previous best BiLSTM model on negation and speculation scope resolution. As discussed previously, BERT was found to implicitly encode various linguistic knowledge after pre-training, and one exploratory work found that this includes negation [124]. However, a concurrent work achieved improvements on the previous SOTA in cue detection and scope resolution on CD-SCO corpus by fine-tuning [125], showing that BERT's knowledge is not complete. A follow up work to [125] found further gains with an MTL setup, which used the same classification head for both negation and speculation resolution by prepending a task marker to the input to tell the model which to perform [126]. The SOTA model at the time of writing is NegBioELECTRA [127]. First the authors "super-tune" a BioELECTRA [128] transformer model on a large number of sentence pairs optimizing for a low cosine similarity between the opposite polarity sentences. They then fine-tune the resulting super-tuned model on the target datasets to perform both cue detection and scope resolution. The resulting model achieves between 97.3-99.5% F1 score on these tasks on both the BioScope and CD-SCO datasets.

#### Contextualizing Events

One of the first systems to tackle the negation and speculation of events, rather than just their scopes, was [57]. Like their approach to EE discussed previously, they extended the patterns of the OpenDMAP system using biomedical ontologies. While their system won first place in BioNLP 2009, their performance on the CC task was quite poor, achieving only 0.03 and 0.09 F1 on negation and speculation, respectively. [129] were one of the first to tackle CC using ML. They utilized SVMs and a number of hand-engineered lexico-syntactic features to detect negated events in the BioNLP 2009 corpus, achieving an F1 score of 0.36, matching the performance of a simultaneously published CRF-based model [130]. While this is a large improvement over the results from [57], it is still low and highlights the difficulty of this task compared to scope resolution.

The EventMine system discussed earlier was extended via additional SVM classifiers to also predict contextual information, called EventMine-MK [131], establishing strong baselines F1 scores of around 82% for both negation and speculation classification on GENIA-MK. A rule-based approach to classifying the factuality of events was developed in [132]. First, the input is mapped to a semantic graph utilizing a hand-written dictionary and rules over the dependency parse. This graph encodes the polarity and uncertainty of words in the input, which forms the basis for the second compositional step that combines these according to the graph structure to determine the factuality of the event, which lies on a scale from from 0 (counterfact) to 1 (fact). This approach lagged somewhat behind the EventMine

system of [131] on GENIA-MK. The research in [39] extends this system with additional compositional rules and applies it to the SemRep Factuality dataset. They also compare both rule- and ML-based approaches to the task of assigning factuality values to semantic predications from SemRep, finding that their system slightly outperformed EventMine-MK on this dataset. The current SOTA on the GENIA-MK and BioNLP 2009 datasets is the UnMine system described in [133], which achieves 0.88 F1 on GENIA-MK and 0.76 F1 and BioNLP, respectively. Their approach uses a random forest classifier with hand-engineered features.

Other than the works highlighted above, research in contextualizing event representations has been sparse. Negation and speculation classification of events was a subtask in both the 2011 and 2013 BioNLP shared tasks, but only 2 teams participated in 2011 and only 3 in 2013. In both cases, the systems were based on SVM pipelines and performances were poor, in the range of 0.2-0.3 F1. The 2022 n2c2 shared task on contextualized medication event extraction in clinical notes was designed to reinvigorate research in this area [134]. All of the top 10 submitted systems were built upon PLMs. These systems ranged from simple fine-tuning of existing PLMs in a pipeline to seq-to-seq models which treated the tasks as a text generation problem. The top performing submission formulated the EE and CC subtasks as a question answering pipeline, in which the prediction from the previous subtask was used as input for the next [135]. Specifically, they added a prompt to the model input depending on the subtask, and then appended the model's prediction for that subtask before moving to the next. For example, given a medication change event, the input for the negation task would be `<sentence text> | Is the medication change negated?`. If the model predicts that the change is negated, the input to the next speculation subtask would be `<sentence text> | Is the medication change negated? | yes | Is the medication change certain?`. Another submission, developed in part by the author, adapted Levitated Markers (discussed in relation to EE above) for CC and was able to automatically identify relevant cue spans for each context dimension without any additional annotation. We discuss this model in more detail in Chapter 5.

### 2.3.2 Datasets

Classifying the negation and speculation status of events can be challenging from a theoretical perspective due to the inherent complexity and occasional idiosyncrasy of their use in natural language. However, they are also challenging tasks from a data perspective. Firstly, EE datasets containing annotations of negation and speculation are less common, and those that do exist are generally smaller than their non-contexual counterparts due to the additional annotation challenge required. Secondly, negated and/or speculative statements are far less frequent than positive and certain statements. Table 2.1 illustrates this, showing the relative number of negated and speculative statements in datasets from the general and biomedical domains in English. These two reasons severely limit the amount of data available for estimating and evaluating NLP models.

| Dataset | Domain | Events | N | Negated (%) | Speculative (%) |
|---|---|---|---|---|---|
| GENIA-MK [136] | Biomedical | ✓ | 36,858 | 2,248 (6.1%) | 2,985 (8.1%) |
| BioScope [7] | | | 20,879 | 2,380 (11.4%) | 3,542 (16.7%) |
| BioNLP 2009 [60] | | ✓ | 11,480 | 723 (6.3%) | 567 (4.9%) |
| SemRep Factuality [39] | | ✓ | 5,328 | 107 (2.0%) | 357 (6.7%) |
| CMED [137] | | ✓ | 1,747 | 38 (2.2%) | 281 (16.1%) |
| SFU Review [138] | General | | 17,263 | 3,125 (18.1%) | 3,919 (22.7%) |
| CD-SCO [121] | | | 5,520 | 1,225 (22.2%) | - |
| ACE-MK [139] | | ✓ | 5,349 | 209 (3.9%) | 1,567 (29.3%) |
| FactBank [140] | | | 2,197 | 53 (2.4%) | 62 (2.8%) |

Table 2.1. Summary statistics of negation and speculation annotations in English biomedical- and general-domain corpora, ordered from largest to smallest.

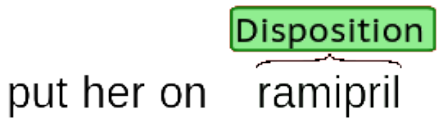
The largest dataset currently available containing both event and context annotations is GENIA-MK, an extension of the GENIA corpus with Meta-Knowledge (MK) [136]. However, it also has the most complex event representations of the datasets in Table 2.1, including nested events and non-contiguous argument spans. While GENIA presents the task as it ought to be solved, with all the complexity present in real scientific text, this complexity can hinder development. To show this complexity, example annotations from CMED, SemRep, and GENIA are given in Figure 2.9. The next largest dataset, BioScope, includes annotations of negation and speculation in biomedical text but does not include any event annotations [7]. The BioNLP 2009 dataset contains simpler event representations than GENIA-MK, but is a third the size and the low percentage of negated and speculative events means that there is little to learn from[4]. Similarly, the SemRep Factuality test corpus and the CMED corpus from the 2022 n2c2 shared task contain few examples. There are a number of other biomedical datasets annotated for negation and uncertainty in languages other than English such as ESSAI/CAS (French) [141], EMR (Chinese) [142], and NUBES (Spanish) [143]. However, the development of multilingual CC systems is a more difficult task and out of scope for this work, so these datasets are not included here.

To further illustrate the issues with existing datasets, we used CMED as a particularly striking example. It contains only 38 negated examples, with 28 in the training split –far too few to learn a generalizable model even with modern pretraining. This leaves only 10 examples for evaluation, which is far too few to be confident in the model's performance.

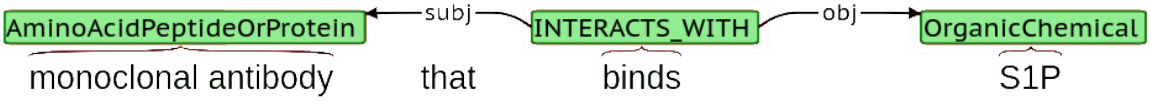
## 2.4 Conclusion

As the sections above show, biomedical EE is a mature subfield of research within NLP whose methods have grown alongside advancements in general NLP an ML. There has been considerable progress since the early days of rule-based systems and classical ML models to modern PLM-based seq-to-seq models. Still, EE is far from a solved problem, as the performance of the current SOTA systems for biomedical EE and RE leave considerable room for improvement. Despite early efforts to include it, the contextualization of events has

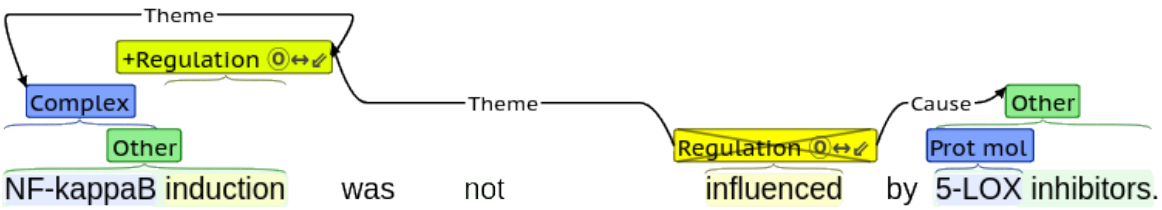
[4]The BioNLP 2011 and 2014 shared tasks also included an EE track annotated with negation and speculation data. However, to the best of our knowledge, these datasets are no longer available to the public so they are not included here.




(a) Example single-span event from the Contextualized Medication Event Dataset (CMED) [137]. "Disposition" refers to a medication change event: in this case, the start of a medication.




(b) Example event from SemMedDB with a *subject-verb-object* structure.




(c) Example event from GENIA, showcasing nested events.

Figure 2.9. Example annotations from the CMED, SemRep, and GENIA datasets.

fallen to the wayside, with current research in negation and speculation processing focused almost entirely on scope resolution which, if the current SOTA systems are any indication, is a solved problem.

Models for CC may be rare, but models that jointly address EE and CC are even rarer, limited to some early rule-based systems and a few submissions from the recent n2c2 shared task on contextualized event extraction. These systems, furthermore, still address EE and CC in a pipeline, rather than fully embrace the necessity of incorporating context into event representations, as argued in Section 1.1. Developing a true joint model for event and context classification requires data, and we argue that the currently available datasets are less than ideal for this due to insufficient examples –especially of the negated and speculative labels– and highly complex event representations (e.g., GENIA-MK), which lend considerable complexity to model development. Therefore, before we discuss potential models for joint EE and CC, Chapter 3 introduces a potential solution to the data problem.

# Chapter 3

# SemMed Context: A large dataset of biomedical events with negation and speculation annotations

Chapter 2 discussed various ways in which existing datasets are ill-suited for developing factualized event extraction (FEE) models. To recap, there are multiple datasets in the biomedical domain that contain both event and negation/speculation annotations, but most of these are relatively small and contain disproportionately few negated and speculative examples. This not only hinders the ability to estimate generalizable models, but also the ability to confidently evaluate model performance on the infrequent labels. Additionally, the largest dataset, GENIA-MK [4], contains many examples but uses a complex event representation that includes non-contiguous argument spans and nested events, which present additional challenges to model design. Specifically, since we argue in Chapter 1 that polarity and uncertainty are necessary dimensionas of the event representation, we are interested in developing multi-task FEE models. A highly complex event representation such as that used by GENIA-MK adds significant complexity to the modeling problem, which is undue in the early stages of model design.

To address these issues, this chapter introduces SemMed Context, a large, weakly-supervised dataset of biomedical events annotated with negation and speculation labels. With over 300,000 examples, the version of SemMed Context introduced here is significantly larger than the other datasets available (the closest is GENIA-MK with 36,000) and, because it is annotated automatically using existing tools, it could be easily expanded to millions of examples given sufficient computing resources. Additionally, the event representations are *subject-predicate-object* triples, a simple yet powerful representation that is common enough to allow model development to draw from the rich literature on RE and EE.

This chapter is organized as follows. The Section 3.1 begins with a discussion of related work on weak supervision for ML dataset construction. It also describes SemRep, an existing NLP tool for biomedical EE that has been previously applied to the entirety of the abstracts in MEDLINE to create the Semantic MEDLINE DataBase (SemMedDB). SemRep Factuality, an extension to SemRep which assigns factuality labels to events is also described. SemMedDB and SemRep Factuality underlie the construction of SemMed Context, and we

describe the five-stage pipeline for generating this dataset in Section 3.2. SemRep and its Factuality extension are far from perfect, however, so this pipeline includes ML filtering models for rectifying incorrect predictions from these tools. Section 3.3 evaluates a simple BERT baseline model for FEE on both SemMed Context and the gold-standard dataset originally used to evaluate SemRep Factuality. Section 3.4 closes the chapter with an alternate view of SemMed Context as a knowledge graph. Entity nodes are merged across claims using UMLS unique identifiers and edges, representing predicates, are merged using Subjective Logic, a framework for mathematical reasoning over uncertainty which allows us to compute consensus over multiple claims extracted from different sources.

## 3.1 Background

This section provides necessary background on existing methods for weak supervision for dataset construction as well as a detailed description of SemRep, SemMedDB, and SemRep Factuality, which form the basis SemMed Context. The necessary background for computing consensus over claims using Subjective Logic is given later in Section 3.4.1.

### 3.1.1 Weak Supervision

All of the existing datasets listed in Section 2.3.2 are constructed using expert human annotators. While this is ideal in theory, in practice it limits the size of the dataset that can be feasibly constructed due to time and resource constraints. This is not a new problem, and a variety of approaches have been introduced for reducing reliance on expert human annotators for dataset construction. These approaches can be categorised into three classes: annotation reduction, distant supervision, and programmatic weak supervision.

#### Annotation reduction

This class of approaches attempts to reduce either the number of annotations required or the complexity of the annotation task. For example, active learning [144] aims to reduce the total annotation load by selecting the most informative examples to annotate using measures of model uncertainty, semi-supervised learning augments a small amount of expertly-annotated data with a larger set of unlabeled data [145], and transfer learning attempts to leverage similar already annotated tasks [146]. There are myriad methods for each of active, semi-supervised, and transfer learning and we refer the curious reader to the citations above as well as [147], [148], and [149] for detailed and up-to-date surveys of each, respectively, reserving space here for a brief summary of previous work using these methods for EE.

Previous research using active learning includes [150], who explore methods for both pipeline and multitask EE models, finding that active learning is most effective for the multi-task model. [151] used active learning to reduce the number of annotations by 92% while losing only 2% accuracy over a fully supervised model on the ACE RE dataset. Similar to the

work proposed here, [152] used active learning to obtain a small number of binary annotations which were used to train an SVM model that filtered the output of SemRep. This not only reduced the amount of annotation required, but also reduced the expertise required of the annotators since the annotation task is a simple binary judgment of whether the output is correct or not, rather than the significantly more complex task of labeling the relations from scratch.

For semi-supervised learning, [153] augment a small number of gold-standard EE examples with varying amounts of data obtained by clustering similar news articles, showing that the addition of the semi-supervised data increases F1 by almost 10% on the ACE dataset when the amount of gold-standard data is small. [154] learn informative features on massive amounts of unlabeled data augmented with a smaller set of labeled data, achieving a performance improvement over fully-supervised methods on BioNLP 2011. In [155] event annotations are generated for large amounts of new data by aligning labeled and unlabeled sentences, reaching SOTA performance on the MLEE biomedical EE dataset [156]. Combining approaches, [157] used active and semi-supervised learning for protein-protein RE, utilizing active learning to discover highly informative examples in the unlabeled data.

Transfer learning was historically defined as the task of repurposing a model trained on one task for a novel, unseen task. More recently, it has been further divided into *few-shot* and *zero-shot* learning. Few-shot learning is most akin to the previous definition of transfer learning: it aims to estimate generalizable models on one dataset and transfer this knowledge to another dataset using minimal supplementary training. A recent example is [158], who develop an encoder-decoder model for EE based on T5 [159] as well as a constrained decoding algorithm for decoding the model output into event structures according to the event schema. Because it uses the schema for decoding, the model can be transferred from one set of event types to another, unseen set. Zero-shot learning is a more constrained version of transfer learning which attempts to solve unseen tasks using existing models without any additional training. [160] perform zero-shot EE by reformulating it as a question answering task, directly evaluating BERT-based models pretrained for textual entailment and question answering on ACE and showing promising results. Another example is [161], who use an existing AMR parser to semantically parse the input which is then mapped to the target event representation by computing a similarity metric between the event mention and an encoding of each event type description, obtained using an PLM.

#### Distant supervision

Going further than mere reduction of annotation load, distant supervision seeks to eliminate the need for expert annotations altogether utilizing simple heuristics, rules, or existing resources to generate very large numbers of labels. Distant supervision has been employed extensively for RE and the predominant method looks for entity pairs in text that are related in an existing knowledge base [162]. While this makes it feasible to extract a very large number of examples, it relies on the assumption that if a pair of entities has a relationship

in the knowledge base, sentences that contain this pair of entities will express that relation. This is a grave assumption which leads to the *wrong labeling problem* where only a subset of sentences containing a pair of entities actually express the target relation, meaning that the labels generated can be noisy. Follow up work generalized this to assume that *at least one* sentence expresses the relation, estimating a graphical model to determine whether a relation is expressed in a given sentence [163]. Still, this approach assumes that only one relation can hold between any two entities, which is often not the case. [164] introduce an additional, corpus level graphical model that is capable of aggregating such relations, outperforming the model by [163]. These graphical models were replaced by a CNN in [165], which further generalised previous research by removing the reliance on hand-engineered features. More recently, [166] perform distantly supervised biomedical RE using the UMLS knowledge base. They obtain graph embeddings from the UMLS which are then used to selectively bias a CNN's internal representations of sentences containing the target subject-object pair. Similarly, [167] combined distant supervision and semi-supervised learning for RE. They augment gold-standard training data with graph embeddings learned from Freebase to align the internal model representations and the target subject-object pair, using a variational autoencoder to learn the sentence representations.

#### Programmatic weak supervision

This "meta-approach" builds off of distant supervision and the graphical model of [163] by aggregating the noisy labels from various labeling functions, leveraging each labeler's different "view" on the data to reduce incorrect labels. More precisely, weak supervision estimates a generative model over the labeling functions in an attempt to recover the true underlying data distribution [168]. In fact, it is shown in [169] that with sufficiently many independent simple labeling functions, weak supervision with generative models outperforms distant supervision and approaches learning from expert annotations. This work introduced Snorkel, a framework for defining labeling functions and learning label aggregation models. Using this tool, [170] constructed a large database of chemical and bacterial reactions directly from raw text, although their evaluation showed a relatively low performance with only 0.55 F1.

There are a few previous works that directly motivate our research. For example, [166] and [171] construct a distantly supervised corpus for RE by identifying pairs of UMLS entities in MEDLINE abstracts that are aligned with the UMLS Metathesaurus. As described above, this is a classic method for distant supervision in RE and while it is able to generate large numbers of examples, it also suffers from considerable label noise. Because our approach utilizes an existing system (SemRep) for data collection, it does not suffer from the wrong labeling problem to the same degree. Another similar work is [172], which improved upon the SemRep filtering model developed in [152] using a pretrained PubMedBERT model. We leverage this filtering model for the construction of SemMed Context both directly –for filtering predications– and indirectly –by retraining it to filter context labels. This work also used the filtered SemRep output to discover novel drug interactions within MEDLINE, illustrating the potential for leveraging existing noisy label sources for knowledge base

construction as we aim to do here.

### 3.1.2 SemRep and SemMedDB

SemRep [56] is a rule-based NLP system developed in the early 2000's at the National Library of Medicine for information extraction from biomedical texts. SemRep is specifically developed for RE and it relies on MetaMap [173] another rule-based tool for named entity recognition (NER) and entity linking to the UMLS Metathesaurus [44]. Given a pair of entities output by MetaMap, SemRep classifies their relationship into one of 30 predicates relating to clinical medicine, substance interactions, medical interventions, and pharmacogenomics, among others[1]. A high-level description of the SemRep pipeline follows and is also illustrated in Figure 3.1. A more detailed discussion can be found in [56]. First, the input text is preprocessed by sentence splitting and tokenization before performing a lexical (lemmatization) and syntactic (POS tagging) analysis. This is then passed to MetaMap, which links noun phrases to entries in the UMLS Metathesaurus. Finally, relations between entity pairs are classified using the SemRep relation ontology and a set of indicator rules, which map certain lexical patterns to predicates.

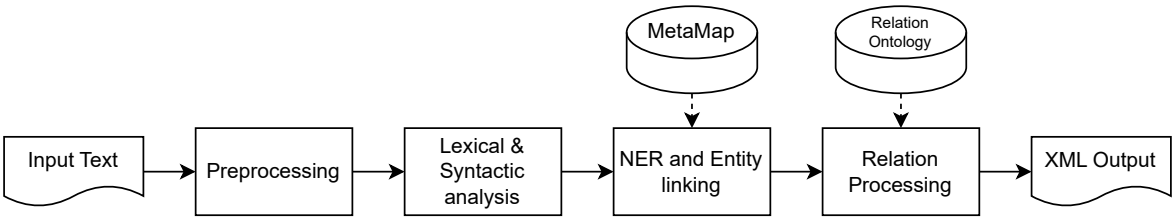



Figure 3.1. An overview of the SemRep pipeline.

SemRep is regularly run over the entirety of the abstracts in MEDLINE to generate a large database of biomedical events, called SemMedDB [175]. Version 40 of SemMedDB, which we use here, was collected from PubMed abstracts published up to 2019 and implemented as a MySQL database. It contains 97,972,561 predications extracted from 18,184,627 different MEDLINE abstracts.

SemRep has been evaluated in multiple scenarios with varying results. For example, [176] recorded F1 scores of 0.58-0.65 for different substance interaction predicates, while [56] report strict and relaxed entity matching F1s of 0.42 and 0.52, respectively, averaged over all predicate types. It is important to note that these results indicate that many of the events in SemMedDB are false positives, and we discuss a solution to this in Section 3.2.1.

### 3.1.3 SemRep Factuality

SemRep Factuality is an extension to SemRep which uses a rule-based approach to assign factuality values to events in SemMedDB [39]. The possible factuality values are *Fact, Probable, Possible, Doubtful* and *Counterfact*[2]. These factuality values can, in fact, be

[1] A complete description of the predicates supported by SemRep can be found in the appendix of [174].

[2] There is also an "Uncommitted" value, but we do not treat of it here as it does not have corresponding negation or certainty values.

decomposed into the polarity and uncertainty annotations developed by [136]. This mapping is given in Table 3.1, reproduced from [39].

| **Factuality** | **Polarity** | **Uncertainty** |
|---|---|---|
| Fact | Positive | L3 (non-speculative) |
| Probable | Positive | L2 (somewhat speculative) |
| Possible | Positive | L1 (considerably speculative) |
| Doubtful | Negative | L1 or L2 |
| Counterfact | Negative | L3 |

Table 3.1. The mapping from the factuality values in SemRep Factuality to polarity and uncertainty values.

SemRep Factuality was shown to obtain good performance on a held out gold-standard in [39]. We reran SemRep Factuality on this gold-standard and report the results in Table 3.2, including the results for the decomposed polarity and uncertainty subtasks.

| | Gold-standard | | | Held-out | | Gold-standard | | | Held-out |
|---|---|---|---|---|---|---|---|---|---|
| Factuality | P | R | F | P | Context | P | R | F1 | P |
| Fact | 0.955 | 0.988 | 0.971 | 0.932 | Negative | 1.000 | 0.537 | 0.698 | 0.345 |
| Probable | 0.653 | 0.701 | 0.676 | 0.848 | Positive | 0.977 | 1.000 | 0.988 | 0.979 |
| Possible | 0.865 | 0.525 | 0.653 | 0.926 | Uncertain | 0.827 | 0.591 | 0.690 | 0.923 |
| Doubtful | 1.000 | 0.333 | 0.500 | 0.506 | Certain | 0.963 | 0.988 | 0.975 | 0.931 |
| Counterfact | 1.000 | 0.571 | 0.727 | 0.356 | | | | | |

Table 3.2. Performance of the rule-based factuality classifier introduced in [39] on their gold-standard test set, as well as the precision of each label manually evaluated on a separate held-out set, randomly selected from SemMedDB and entirely distinct from the gold-standard test set.

In addition to the relatively low performance of SemRep on predicates discussed above, we anecdotally noticed that in practice SemRep Factuality seemed to perform poorly when run on predications randomly selected from SemMedDB. We therefore performed a secondary evaluation of SemRep Factuality to further assess its performance. This evaluation consisted of a manual review of the output of SemRep Factuality for 3,000 predications randomly selected from SemMedDB stratified by the relevant task label. We only marked whether each prediction was correct or not, meaning that the evaluation measured the precision of each label only, since we do not have the gold-standard labels to compute recall. The results of this manual evaluation are given in the Held-out columns of Table 3.2, which show that SemRep Factuality indeed performs differently that its results on the gold-standard suggest, especially on the Negative label. The high precisions of *Probable* and *Possible* compared to the low precisions of *Doubtful* and *Counterfact* suggest that SemRep Factuality performs poorly on negation processing. To confirm this, we decomposed the Factuality labels into their polarity and uncertainty components and evaluated the precision of each. The results, given in the right-hand Held-out column of Table 3.2, confirm our suspicions: the negative label obtains an precision of only 0.345, while the positive performs well at 0.979. Additionally, we note that the uncertain label performs better in our evaluation than on the gold-standard dataset, achieving an precision of 0.923.

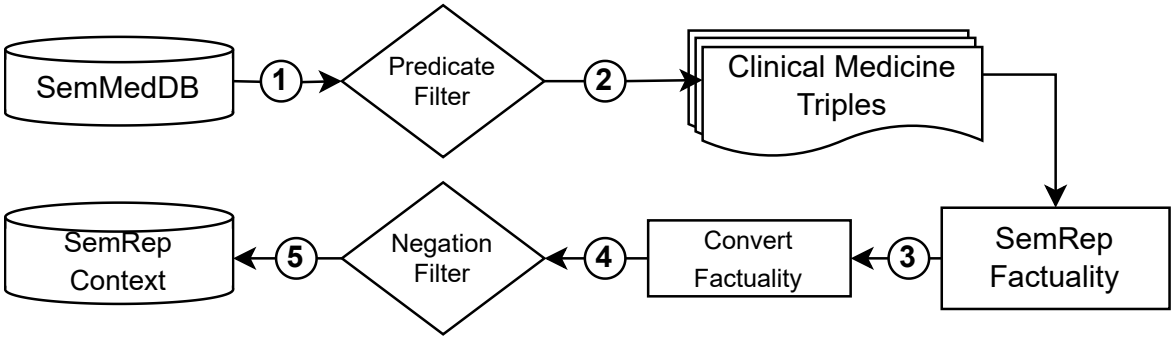



Figure 3.2. An overview of the SemMed Context construction process, broken up into 5 steps.

## 3.2 SemMed Context

Our proposal utilizes SemMedDB and SemRep Factuality to automatically generate a large dataset of biomedical events annotated with polarity and uncertainty labels in order to provide a large corpus for the development and evaluation of models for FEE.

### 3.2.1 Construction

Construction of SemMed Context can be broken up into 5 stages, illustrated in Figure 3.2 and described in detail in the following sections: Step 1 queries predications from SemMedDB. Step 2 applies a predicate filtering model to remove examples incorrectly extracted by SemRep. Step 3 passes these filtered predications to SemRep Factuality, which assigns factuality labels. Step 4 converts these factuality labels to polarity and uncertainty labels per Table 3.1. Step 5 uses another filtering model to reannotate incorrectly negated predications.

#### 1. Extracting predications from SemMedDB

The first step in building SemMedContext is to extract events of interest from SemMedDB. Events in SemMedDB are (subject, predicate, object) triples, where the predicate is one of 30 types [174]. In order to make modeling more tractable, we first limit the number of events extracted to 500,000 as well as follow previous work and extract events containing predicates related to clinical medicine and substance interactions [152], which have been used for drug re-purposing [177]. These predicates, along with the count of each obtained from our query, are given in Table 3.3.

Ideally, our final dataset would have a roughly even number of positive/negated and certain/uncertain examples for each predicate type, yet when querying SemMedDB we do not know whether the predication is negated or uncertain, although we do know that these labels are less frequent than their positive and certain counterparts. In an effort to bolster the numbers of these examples, our query prefers examples whose sentence contains a negation or speculation cue extracted from the set of cues annotated in the SFU Review corpus [178]. Our query always keeps a predicate if its sentence contains a cue, and keeps predicates without cues with a 10% probability. To evaluate the effect of cue matching, we constructed two

| Predicate | N | % | Predicate | N | % |
|---|---|---|---|---|---|
| TREATS | 99373 | 19.87 | AUGMENTS | 19532 | 3.91 |
| AFFECTS | 74463 | 14.89 | DISRUPTS | 17586 | 3.52 |
| INTERACTS_WITH | 59579 | 11.92 | PRODUCES | 15732 | 3.15 |
| COEXISTS_WITH | 55693 | 11.14 | PREDISPOSES | 9929 | 1.99 |
| CAUSES | 42226 | 8.45 | PREVENTS | 7776 | 1.56 |
| STIMULATES | 35179 | 7.04 | MANIFESTATION_OF | 1306 | 0.26 |
| ASSOCIATED_WITH | 34004 | 6.80 | COMPLICATES | 699 | 0.14 |
| INHIBITS | 26923 | 5.38 | **Total** | **500,000** | **100** |

Table 3.3. Number and percentage of each predicate extracted from SemMedDB.

smaller datasets of 10,000 examples with and without cue matching, respectively. Before negation filtering (step 5), cue matching resulted in an increase of over 1% (0.85% to 2.03%) of examples with a Negated label, although after filtering the increase dropped (from 0.44% to 0.73%). For uncertain examples, we saw an increase of almost 12% (9.12% to 21.1%). This suggests that cue matching in this step provides significant increase, even if the total number of negated examples extracted is still low.

#### 2. Filtering incorrect predications with machine learning

Previous work found that the precision of the events extracted by SemRep and incorporated into SemMedDB is low, in the range of 0.42-0.69 depending on the evaluation setup [56], [176]. This low precision has been addressed in previous work by post-hoc filtering of incorrect examples with machine learning models based on SVMs [152] and transformers [172], [177], the latter of which achieved an accuracy of 0.91 by fine-tuning PubMedBERT [179]. The input to this model is the full sentence concatenated with the (subject, predicate, object) tuple output by SemRep. A single fully-connected layer transforms the pooled output of the PubMedBERT model into a binary prediction regarding the correctness of the triple given the sentence. We utilize the publicly available trained model checkpoint for filtering SemMed Context[3]. After filtering, SemMed Context contains 310,810 events, 62% of the original 500,000. We also remove examples with the *COMPLICATES* and *MANIFESTATION_OF* predicates due to few examples, resulting in a total of **309,609** examples. The statistics of the dataset after filtering is given in Table 3.4.

| Predicate | N | % | Predicate | N | % |
|---|---|---|---|---|---|
| TREATS | 71165 | 22.90 | INHIBITS | 9486 | 3.05 |
| AFFECTS | 45867 | 14.76 | DISRUPTS | 8720 | 2.81 |
| COEXISTS_WITH | 42677 | 13.73 | PRODUCES | 7944 | 2.56 |
| CAUSES | 33990 | 10.94 | PREDISPOSES | 7873 | 2.53 |
| ASSOCIATED_WITH | 26777 | 8.62 | PREVENTS | 6445 | 2.07 |
| INTERACTS_WITH | 26496 | 8.52 | **Total** | **309,609** | **100** |
| STIMULATES | 11738 | 3.78 | COMPLICATES | 682 | 0.22 |
| AUGMENTS | 10431 | 3.36 | MANIFESTATION_OF | 519 | 0.17 |

Table 3.4. Number and percentage of each predicate extracted from SemMedDB after filtering.

[3]Downloaded from `https://github.com/kilicogluh/lbd-covid/tree/master/filtering`

#### 3. Running SemRep Factuality

We then run SemRep Factuality on the examples from step 2. Because it does not map to polarity/uncertainty values, we remove 6,714 examples given the Uncommitted label resulting in 302,895 total predications.

#### 4. Obtaining negation and uncertainty annotations

We further condense the certainty levels L2 and L1 (cf. Table 3.1) into a single "uncertain" label and treat L3 into "certain", following previous work that showed the unreliability of annotations differentiating between L1 and L2 [180].

| Predicate | N (%) | Negated (%) | Uncertain (%) |
|---|---|---|---|
| TREATS | 71,165 (22.90) | 1,417 (1.99) | 4,411 (6.20) |
| AFFECTS | 45,867 (14.76) | 1,724 (3.76) | 7,031 (15.33) |
| CAUSES | 33,990 (10.94) | 828 (2.44) | 4,836 (14.23) |
| COEXISTS_WITH | 42,677 (13.73) | 491 (1.15) | 1,619 (3.79) |
| ASSOCIATED_WITH | 26,777 (8.62) | 543 (2.03) | 1,684 (6.29) |
| INTERACTS_WITH | 26,496 (8.52) | 841 (3.17) | 3,141 (11.85) |
| STIMULATES | 11,738 (3.78) | 309 (2.63) | 1,151 (9.81) |
| AUGMENTS | 10,431 (3.36) | 281 (2.69) | 1,297 (12.43) |
| INHIBITS | 9,486 (3.05) | 291 (3.07) | 926 (9.76) |
| DISRUPTS | 8,720 (2.81) | 255 (2.92) | 1,041 (11.94) |
| PRODUCES | 7,944 (2.56) | 284 (3.58) | 577 (7.26) |
| PREDISPOSES | 7,873 (2.53) | 274 (3.48) | 1,628 (20.68) |
| PREVENTS | 6,445 (2.07) | 248 (3.85) | 1,266 (19.64) |
| **Total** | **302,895 (100)** | **7,786 (2.57)** | **30,608 (10.11)** |

Table 3.5. Number and percentage of each predicate extracted from SemMedDB after filtering, along with the number of Negated and Uncertain examples for each. The percentages for each predicate are computed relative to the total number of examples, while the percentages of Negated and Uncertain examples are computed relative to the number of the corresponding predicate.

#### 5. Filtering incorrect negation annotations

We annotated an independent set of 1,111 SemMedDB events that were assigned the negative polarity label by SemRep Factuality. This revealed that only 35% of these were correctly labeled as negated. We therefore fine-tuned the same PubMedBERT filtering model used above on these 1,111 events. This model achieved a precision of 0.89 on a small validation set and 0.75 in the final SemRep Context version, as shown in Table 3.7. A manual evaluation of 200 events assigned the uncertain label showed that SemRep Factuality achieved a precision of 0.92 on the uncertainty task (cf. Table 3.2). We therefore did not deem it crucial to train a model to filter incorrect uncertain examples.

As Table 3.6 shows, negation filtering reannotates over half of the previously negative examples, resulting in only 1% of the final dataset being negative.

| Predicate | N (%) | Negated (%) | Uncertain (%) |
|---|---|---|---|
| TREATS | 70,183 (23.17) | 512 (0.73) | 4,411 (6.20) |
| AFFECTS | 43,698 (14.43) | 529 (1.21) | 7,031 (15.33) |
| CAUSES | 33,467 (11.05) | 201 (0.60) | 4,836 (14.23) |
| COEXISTS_WITH | 42,412 (14.00) | 150 (0.35) | 1,619 (3.79) |
| ASSOCIATED_WITH | 26,194 (8.65) | 242 (0.92) | 1,684 (6.29) |
| INTERACTS_WITH | 25,196 (8.32) | 395 (1.57) | 3,141 (11.85) |
| STIMULATES | 11,629 (3.84) | 168 (1.44) | 1,151 (9.81) |
| AUGMENTS | 10,314 (3.41) | 141 (1.37) | 1,297 (12.43) |
| INHIBITS | 9,365 (3.09) | 187 (2.00) | 926 (9.76) |
| DISRUPTS | 8,644 (2.85) | 151 (1.75) | 1,041 (11.94) |
| PRODUCES | 7,868 (2.59) | 132 (1.68) | 577 (7.26) |
| PREDISPOSES | 7,592 (2.51) | 89 (1.17) | 1,628 (20.68) |
| PREVENTS | 6,333 (2.09) | 152 (2.40) | 1,266 (19.64) |
| **Total** | **302,865 (100)** | **3,049 (1.01)** | **30,608 (10.11)** |

Table 3.6. Number and percentage of each predicate extracted from SemMedDB after negation filtering, along with the number of Negated and Uncertain examples for each. The percentages for each predicate are computed relative to the total number of examples, while the percentages of Negated and Uncertain examples are computed relative to the number of the corresponding predicate.

### 3.2.2 Evaluation of the filtering pipeline

We manually re-evaluated the precision of the negative polarity, counterfact, and doubtful labels after filtering on a held out set. This is shown in the Held-out columns of Table 3.7 alongside the previous evaluation results for comparison.

| | Gold-standard | | | Held-out | |
|---|---|---|---|---|---|
| | P | R | F | P | P Filtered |
| Negative | 1.000 | 0.537 | 0.698 | 0.345 | 0.750 |
| Positive | 0.977 | 1.000 | 0.988 | 0.979 | - |
| Uncertain | 0.827 | 0.591 | 0.690 | 0.923 | - |
| Certain | 0.963 | 0.988 | 0.975 | 0.931 | - |
| Fact | 0.955 | 0.988 | 0.971 | 0.932 | - |
| Probable | 0.653 | 0.701 | 0.676 | 0.848 | - |
| Possible | 0.865 | 0.525 | 0.653 | 0.926 | - |
| Doubtful | 1.000 | 0.333 | 0.500 | 0.506 | 0.586 |
| Counterfact | 1.000 | 0.571 | 0.727 | 0.356 | 0.739 |

Table 3.7. Precision, Recall, and F1 score of SemRep Factuality on the gold-standard test set. Precision of each negation, and uncertainty, and factuality label assigned by SemRep Factuality before (P) and after (P Filtered) filtering, compared to the gold-standard results.

As Table 3.7 shows, the performance of SemRep Factuality on the gold-standard data does not match its performance on predications randomly selected from SemMedDB, although our filtering step vastly increases the precision of the negative label.

### 3.2.3 Dataset Description

We further split SemMed Context into training, validation, and testing sets for model estimation, stratified on the predicate labels. Statistics of the polarity, uncertainty, and factuality labels are given in Table 3.8 and compared to the SemRep Factuality dataset used in [39].

| | SemMed Context | | | SemRep Factuality | |
|---|---|---|---|---|---|
| | Train (%) | Val (%) | Test (%) | Train (%) | Test (%) |
| Predications | 242,016 | 30,380 | 30,469 | 3,028 | 2,127 |
| Negative | 2,452 (1.0%) | 305 (1.0%) | 292 (1.0%) | 65 (2.1%) | 42 (2.0%) |
| Uncertain | 24,257 (10.2%) | 3,140 (10.3%) | 3,211 (10.5%) | 217 (7.2%) | 134 (6.3%) |
| Fact | 215,447 (89.0%) | 26,953 (88.7%) | 26,988 (88.6%) | 2,754 (91.0%) | 1,958 (92.1%) |
| Probable | 16,403 (6.8%) | 2,106 (7.0%) | 2,191 (7.2%) | 143 (4.7%) | 67 (3.1%) |
| Possible | 7,714 (3.2%) | 1,016 (3.3%) | 998 (3.2%) | 66 (2.2%) | 61 (2.9%) |
| Doubtful | 140 (0.1%) | 18 (0.1%) | 22 (0.1%) | 8 (0.3%) | 6 (0.3%) |
| Counterfact | 2,312 (0.9%) | 287 (0.9%) | 270 (0.9%) | 57 (1.9%) | 35 (1.6%) |

Table 3.8. Comparison of the SemRep Factuality and SemMed Context splits. Note that SemRep Factuality does not have a validation split. We omit the Conditional and Uncommitted Factuality examples from SemRep Factuality due to low counts and not having polarity/uncertainty labels, respectively.

## 3.3 Baseline Model

We here evaluate baseline models for polarity, uncertainty, predicate, and factuality classification on SemMed Context.

### 3.3.1 Model Description and Evaluation Setup

We estimate single-task models for predicate, negation, and uncertainty by fine-tuning BERT.[4] In all models, a linear layer transforms the sentence representation stored in the `[CLS]` token into predictions. As shown in Table 3.8 and Table 3.6, there are large differences in label frequencies within each task, although these frequencies are consistent across the train, validation, and test splits. To address these imbalances, we train using cross entropy loss, weighting the loss of each example inverse to its label's proportion in the training data.

While model estimation always occurs on the SemRep Context training split, we provide two evaluation scenarios. In the first, we report performance on the SemRep Context validation and test sets. In the second, we report performance on the gold-standard SemRep Factuality test split from [39]. For the polarity and uncertainty tasks we report precision, recall, and F1 scores for each label, while we report macro averaged metrics for the Predicate task. We report these metrics for each factuality value to compare with the results published in [39].

### 3.3.2 Results

The baselines performs notably worse on the SemRep Factuality test set compared to SemMed Context, especially on the Negated and Uncertain labels. We note that the evaluated precision of the negative label is 75% (cf. Table 3.7). Assuming that the drop in performance from SemMed Context to SemRep Factuality is caused entirely by the 25% error rate of the filtered negative predictions, we expect a drop in precision, recall, and F1 of 25% as well.

[4] Interestingly, preliminary experiments showed that the general-domain BERT embeddings (`bert-base-uncased`) outperformed PubMedBERT on these tasks. While a more complete evaluation of different pre-trained BERT embeddings would be ideal, we leave this to future work due to limited computational resources.

| | | Polarity | | | | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Negative | | | Positive | | | Macro Avg. | | |
| | Split | P | R | F | P | R | F | P | R | F |
| SemMed Context | val | 0.479 | 0.492 | 0.485 | 0.995 | 0.995 | 0.995 | 0.792 | 0.781 | 0.785 |
| | test | 0.462 | 0.517 | 0.488 | 0.995 | 0.994 | 0.995 | 0.795 | 0.778 | 0.785 |
| SemRep Fact | test | 0.273 | 0.220 | 0.243 | 0.982 | 0.997 | 0.989 | 0.700 | 0.619 | 0.637 |

| | | Uncertainty | | | | | |
|---|---|---|---|---|---|---|---|
| | | Uncertain | | | Certain | | |
| | Split | P | R | F | P | R | F |
| SemMed Context | val | 0.719 | 0.818 | 0.765 | 0.979 | 0.963 | 0.971 |
| | test | 0.720 | 0.821 | 0.767 | 0.979 | 0.962 | 0.970 |
| SemRep Fact | test | 0.315 | 0.677 | 0.430 | 0.977 | 0.904 | 0.939 |

Table 3.9. Results of the baseline BERT model trained on SemMed Context and evaluated on both SemMed Context and the SemRep Factuality test set. For the predicate task the model was evaluated on the subset of the SemRep Factuality test set that contained the predicates used in SemMed Context. For the polarity model, we weight the loss function according to the proportion of classes in the training data to account for label imbalance.

| | Kilicoglu et al. 2017 | | | BERT Factuality | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| Fact | 0.955 | 0.988 | 0.971 | 0.962 | 0.905 | 0.933 |
| Probable | 0.653 | 0.701 | 0.676 | 0.236 | 0.651 | 0.346 |
| Possible | 0.865 | 0.525 | 0.653 | 0.383 | 0.590 | 0.465 |
| Doubtful | 1.000 | 0.333 | 0.500 | 0.667 | 0.333 | 0.444 |
| Counterfact | 1.000 | 0.571 | 0.727 | 0.500 | 0.143 | 0.222 |
| Macro Avg. | 0.895 | 0.624 | 0.705 | 0.549 | 0.524 | 0.482 |

Table 3.10. The baseline BERT models compared to the best performing model from [39] on the SemRep Factuality test set. BERT Factuality: The baseline model fine-tuned on Factuality annotations. BERT Neg+Cert: The baseline model fine-tuned on Negation and Certainty annotations, using the equivalences in Table 3.1 to map predictions back to Factuality.

Thus we would expect the SemRep Fact test set metrics to be 0.347, 0.388, and 0.366, respectively, higher than the actual results of 0.273, 0.220, and 0.243. This suggests that the drop in performance from SemMed Context to SemRep Fact is partly due to a data mismatch. This is supported by the drop in performance of around 44% on the uncertain label, despite our manual evaluation showing that SemRep has an precision of 92% on this label (cf. Table 3.7). Investigating further, we inverted our evalutation setup, training the model on the SemRep Factuality train split and evaluating it on SemMed Context test split. The results in Table 3.11 further support a data mismatch: the model performs notably worse on the SemRep Factuality test split than the validation split, and performance degrades even further when evaluated on SemMed Context.

The results in this section show that 1) despite being weakly supervised, SemMed Context is a suitable training dataset for FEE, as shown by the results in Table 3.7; 2) there is room for modeling improvement, as the baseline model struggles on the lower frequency labels for all tasks, but especially polarity where the model only obtains 0.485 F1 on the negative label.

| | | Polarity | | | | | |
|---|---|---|---|---|---|---|---|
| | | Negative | | | Positive | | |
| | Split | P | R | F | P | R | F |
| SemRep Fact | val | 0.666 | 0.750 | 0.706 | 0.984 | 0.976 | 0.980 |
| | test | 0.519 | 0.452 | 0.483 | 0.980 | 0.985 | 0.982 |
| SemMed Context | test | 0.069 | 0.603 | 0.123 | 0.996 | 0.921 | 0.957 |

| | | Uncertainty | | | | | |
|---|---|---|---|---|---|---|---|
| | | Uncertain | | | Certain | | |
| | Split | P | R | F | P | R | F |
| SemRep Fact | val | 0.654 | 0.850 | 0.739 | 0.972 | 0.920 | 0.946 |
| | test | 0.506 | 0.669 | 0.576 | 0.943 | 0.893 | 0.917 |
| SemMed Context | test | 0.348 | 0.610 | 0.443 | 0.950 | 0.866 | 0.906 |

Table 3.11. Results of the baseline BERT model trained on SemRep Factuality and evaluated on both SemRep Factuality and the SemMed Context test set. Note that for model development we chose a stratified subset of examples from the SemRep Factuality train set to use for validation. We also only trained and evaluated the model on the subset of SemRep Factuality that contained the predicates used in SemMed Context. For the Negation model, we weight the loss function according to the proportion of classes in the training data to account for label imbalance. There are too few examples in the train set to estimate a Predicate model, so we do not report those results here.

## 3.4 Computing Consensus over Claims

In this section we discuss preliminary work on the applications of modeling event factuality using polarity and uncertainty. Specifically, we show that event factuality can be represented using Subjective Logic (SL), an extension of probabilistic logic [40]. Additionally, we discuss how to combine and ultimately reason over events using SL operators.

### 3.4.1 Background: Subjective Logic

Subjective Logic (SL) is a mathematical framework for reasoning over uncertainty. It extends the binary truth of formal logic into probabilistic logic which treats premises as *beliefs* belonging to a subject (e.g., a person, a newspaper article, an unreliable mechanical sensor). These *subjective opinions* have scalar truth values that also encode the uncertainty of the subject regarding the premise. Formally, given a premise $x$, SL defines a binomial opinion regarding the truth of $x$ as

$$\omega_x = (b_x, d_x, u_x, a_x) \tag{3.1}$$

where $b_x$ is the belief in the truth of $x$, $d_x$ is the corresponding disbelief, $u_x$ is the subject's uncertainty in the belief, and $a_x$ is a prior probability of $x$ being true (in practice it is often set to an uninformative 0.5). The values of these parameters must satisfy the constraints $b_x, d_x, u_x, a_x \in [0, 1]$ and $b_x + d_x + u_x = 1$. Table 3.12 describes the possible types of opinions according to the parameter values.

Subjective opinions can be represented in two ways: as points on a simplex with $b$, $d$, $u$ vertices or as Beta probability density functions (PDFs) via the following parameter transfor-

| $b$ | $d$ | $u$ | **Interpretation** |
|---|---|---|---|
| 1 | 0 | 0 | Absolute opinion, equivalent to binary True. |
| 0 | 1 | 0 | Absolute opinion, equivalent to binary False. |
| $< 1$ | $< 1$ | 0 | Dogmatic opinion, equivalent to traditional probability. |
| $< 1$ | $< 1$ | $< 1$ | Uncertain opinion. |
| 0 | 0 | 1 | Vacuous opinion with complete uncertainty. |

Table 3.12. Classification of subjective opinions in Subjective Logic according to the values of the $b$, $d$, and $u$ parameters.

mation:

$$\text{Beta}_x(\alpha, \beta) = \begin{cases} \alpha = \frac{2b_x}{u_x} + 2a_x \\ \beta = \frac{2d_x}{u_x} + 2(1 - a_x) \end{cases} \tag{3.2}$$

This representation as a probability distribution is what connects SL to traditional probability theory. For example, the probability of an event is the same as the expectation of the Beta PDF, which can computed as $P_x = b_x + a_x u_x$. This connection goes deeper as well, as SL defines a number of operators on subjective opinions that form a common ground between formal logical operators (e.g., `AND`, `OR`, `NOT`) and probability theory (e.g., multiplication, addition, complement). For example, addition is defined as

$$\omega_{(x_1 \cup x_2)} = \omega_{x_1} + \omega_{x_2} := \begin{cases} b_{x_1} \cup b_{x_2} = b_{x_1} + b_{x_2} \\ d_{x_1} \cup d_{x_2} = \frac{a_{x_1}(d_{x_1} - b_{x_2}) + a_{x_2}(d_{x_2} - b_{x_0})}{a_{x_1} + a_{x_2}} \\ u_{x_1} \cup u_{x_2} = \frac{a_{x_1} u_{x_1} + a_{x_2} u_{x_2}}{a_{x_1} + a_{x_2}} \\ a_{x_1} \cup u_{x_2} = a_{x_1} + a_{x_2} \end{cases} \tag{3.3}$$

and it can be shown that the addition of subjective opinions is a generalization of the addition of probabilities. That is, $P_{(x_1 \cup x_2)} = P_{x_1} + P_{x_2}$.

Because SL includes the uncertainty dimension, it defines additional operators over subjective opinions that leverage uncertainty. One example, and the operator we will utilize later for computing consensus, is *cumulative fusion*, which essentially adds the evidence from two opinions to obtain a single meta-opinion representing consensus. Given two subjective opinions $\omega^A$ and $\omega^B$ from two subjects $A$ and $B$ regarding a single premise, cumulative fusion is defined as

$$\omega^{(A \diamond B)} = \omega^A \oplus \omega^B := \begin{cases} b^{(A \diamond B)} = \frac{b^A u^B + b^B u^A}{u^A + u^B - u^A u^B} & \\ d^{(A \diamond B)} = 1 - u^{(A \diamond B)} & \\ u^{(A \diamond B)} = \frac{u^A u^B}{u^A + u^B - u^A u^B} & \\ a^{(A \diamond B)} = \frac{a^A u^B + a^B u^A - (a^A + a^B) u^A u^B}{u^A + u^B - 2u^A u^B} & \text{if } \; u^A \neq 1 \wedge u^B \neq 1 \\ a^{(A \diamond B)} = \frac{a^A + a^B}{2} & \text{if } \; u^A = u^B = 1 \end{cases} \tag{3.4}$$

There are two versions of cumulative fusion, depending on the type of opinions that are being merged. Equation (3.4) defines aleatoric cumulative fusion, which applies when the premise is a frequentist process where additional observations *reduce* uncertainty. Epistemic cumulative fusion, on the other hand, applies to non-frequentist processes when additional observations do not necessarily reduce uncertainty. To illustrate the difference, let us assume that two different people are counting the number of heads of a biased coin through a noisy video monitor (adding a degree of uncertainty to their observations). If the sets of flips observed by each person do not overlap we would merge their observations using aleatoric cumulative fusion to estimate the true bias of the coin. If they do overlap (i.e., their observations are not independent) then we would use epistemic cumulative fusion. Epistemic cumulative fusion is denoted $\ddot{\oplus}$ and defined as in Equation (3.4) with the additional step of maximizing the uncertainty $u$ of the resulting distribution and reducing the belief $b$ and disbelief $d$ such that the expectation of the distribution remains the same. Examples of aleatoric and epistemic cumulative fusion between two subjective opinions are illustrated in Figure 3.3.

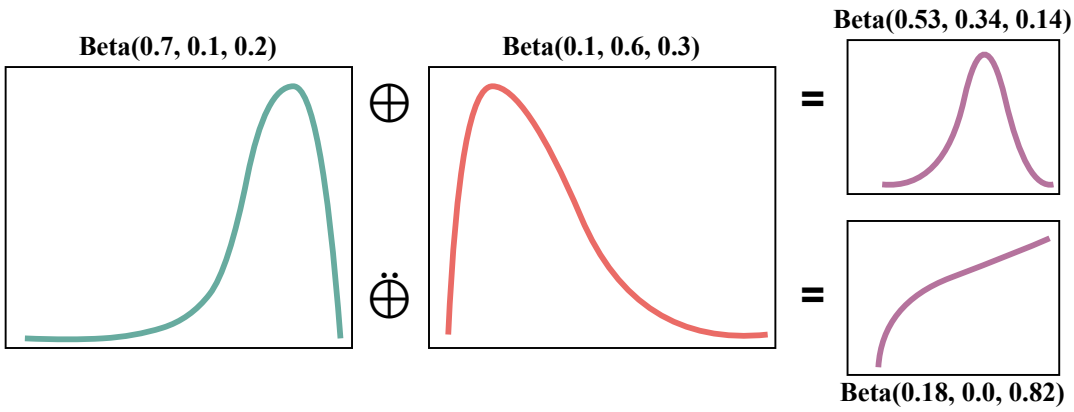



Figure 3.3. Examples of aleatoric $\oplus$ and epistemic $\ddot{\oplus}$ cumulative fusion of two subjective opinions depicted as Beta PDFs.

### 3.4.2 SemMed Context as a Knowledge Graph

As described above, each claim in SemMed Context is assigned a polarity and uncertainty value derived from the factuality label predicted by SemRep Factuality. A trivial mapping allows us to represent the factuality of each claim using SL, as the polarity dimension maps to the $b$ and $d$ parameters and the uncertainty dimension to $u$. We hand-defined the parameter values of subjective opinions for each factuality value, such that the modes of the Beta PDFs evenly spread between $[0, 1]$. The parameter values and depictions of each factuality value are given in Figure 3.4.

Using this mapping we represented the factuality of each claim in SemMed Context as a subjective opinion. We then used epistemic cumulative fusion to merge claims that dealt with the same subject, predicate, and object, treating each event as a unit of evidence for (or against) the truth of the claim. From the 301,320 total claims in SemMed Context we extracted 227,789 unique triples. Of these, 26,039 have more than 1 piece of supporting

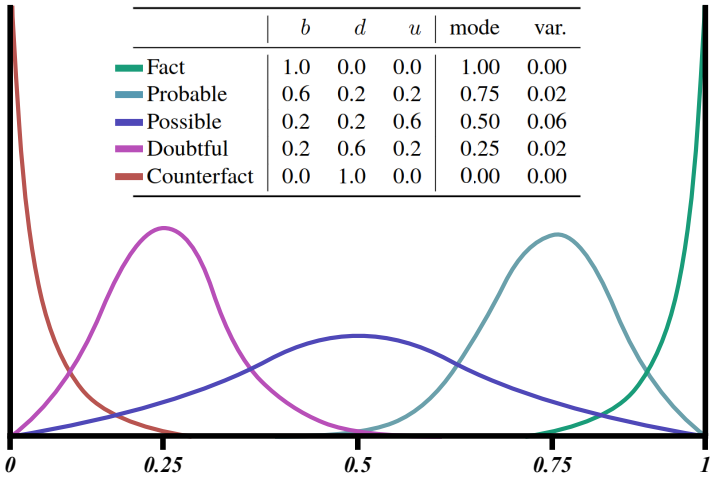



Figure 3.4. Representation of the factuality values in SemRep Factuality as Beta distributions. The negation dimension is represented by the $x$-axis and the certainty dimension by the distribution variance. The mapping of factuality values to scalars is from [39]. Note that while the Fact and Counterfact distributions have 0 variance, we add variance to the figure for ease of visualization.

evidence and cover 9,207 different entities. There are 218,540 unique evidence PMIDs, with the majority of PMIDs supporting only 1 event, although 54,781 PMIDs support $> 1$ claim with the maximum being 26. Summary statistics of the number of each factuality value the knowledge graph are given in Table 3.13.

| Support | N | Fact | Probable | Possible | Doubtful | Counterfact |
|---|---|---|---|---|---|---|
| $\geq 0$ | 227,786 | 198,061 | 16,198 | 7,832 | 367 | 5,328 |
| $\geq 2$ | 26,039 | 24,052 | 1,035 | 775 | 23 | 154 |

Table 3.13. Count of each Factuality value in SemMed Context after consensus, computed using thresholds of the belief value $b$.

We implement the knowledge graph in Neo4j in which entity nodes are keyed by UMLS CUIs and edges by predicate type. Edges are also annotated with their consensus belief and the PMIDs of all supporting claims.

A subgraph of SemMed Context showcasing relationships between the Epstein-Barr virus and different diseases is given in Figure 3.5. In this subgraph, we see a doubtful `CAUSES` relationship between Epstein-Barr Virus and Breast Cancer. This edge is supported by two claims from [181] and [182]. Specifically, the evidence sentences are

- [181]: "Considering the study's statistical power, these results do not support the involvement of EBV and MMTV-LV in the etiology of breast cancer."
- [182]: "These and additional data provide substantial, but not conclusive, evidence that human papillomavirus, the mouse mammary tumor virus and Epstein-Barr virus may have a role in the etiology of human breast cancer."

We can see that the claims are contradictory, and that the negative claim from [181] is more certain than the speculative claim in [182], resulting in a Doubtful relationship $\omega_x = (0.23, 0.59, 0.18, 0.5)$.

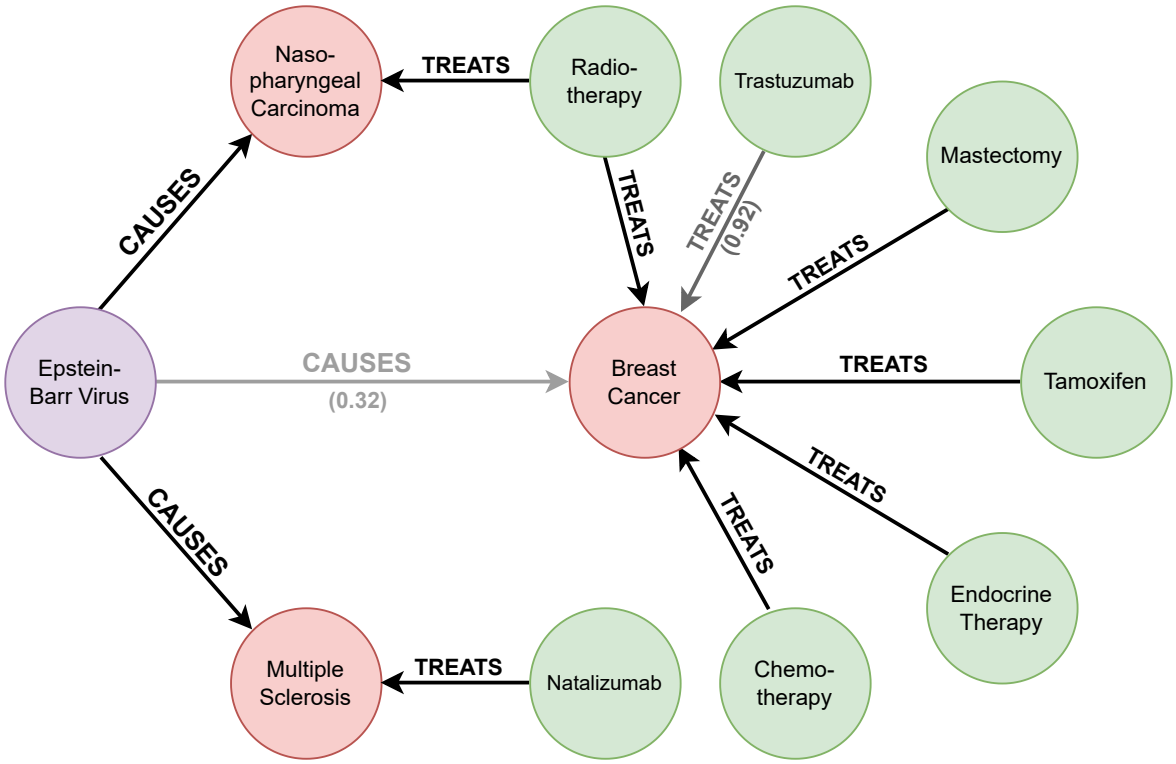



Figure 3.5. A subgraph of SemMed Context showing the relationships between Epstein-Barr Virus, three diseases, and their respective treatments, all with $\geq 2$ supporting claims. The consensus belief of each relationship is represented by the darkness of the arrow color and we label relationships with less than a Fact label with their corresponding belief value in parentheses.

### 3.4.3 Preliminary Work: Embracing Disagreement

Previous research has shown that while annotator agreement on polarity is quite high, there is often considerable disagreement regarding the degree of uncertainty of a claim [133]. This disagreement was addressed by consolidating uncertainty annotations into binary ones, and we followed suit in this work. While this effectively reduced disagreement, it did also reduce the granularity of the uncertainty annotations. On the other hand, we may also consider that such disagreement is inherent in the task and embrace it. Embracing, rather than removing, annotator disagreement is the focus of a new approach to learning from labeled data, called *data perspectivism* [37]. Under data perspectivism, multiple annotators provide labels for each example in a dataset. These label sets are then used to create target distributions over labels for each example, rather than using a single annotation as a gold-standard. This means that where there is no disagreement between annotators, the result is comparable to a gold label. There are a number of methods for computing target distributions from individual annotations, but Subjective Logic as a framework for representing beliefs and operations for computing consensus is a natural choice for this task. Unfortunately, there are no datasets with crowd-sourced labels for textual uncertainty. Still, Appendix A describes a method for constructing target distributions by aggregating labels using Subjective Logic and evaluates this method on synthetic data.

## 3.5 Limitations and Future Work

The version of SemMed Context constructed here utilizes only a fraction of the many millions of predications available in SemMedDB, and is restricted to a subset of the possible predicate types. Future work should expand SemMed Context to process the entirety of SemMedDB. Additionally, we noted discrepancies between model performance on the SemRep Factuality gold-standard and SemMed Context, as our baseline model failed to generalize from SemMed Context to SemRep Factuality. Our hypothesis is that the SemRep Factuality test split is not entirely representative of the task, but additional work is required to identify precisely where. Finally, despite its size and our methods for increasing their numbers, SemMed Context contains relatively few negative and uncertain examples. Neither does it included an "Unrelated" predicate. While such a predicate does not exist within SemMedDB, it is important to train a model to recognise non-related entity pairs in order to function on real-world data.

It is also possible to further reduce label noise by employing programmatic weak supervision. Potential labeling functions that could be employed to either reduce the reliance on human annotation or improve the accuracy of the predictions might be:

- **Crowd-sourcing:** This aligns closely with our proposal to treat labels as "beliefs", since the annotations are supplied by non-expert human annotators.
- **Large Language Models:** Modern LLMs can be queried zero-shot to provide additional views on the data. We performed a preliminary investigation of two LLMs for classifying the polarity and uncertainty between entities. Details of this evaluation are provided in Appendix C.
- **Scope Resolution Models:** As discussed in Chapter 2, SOTA performance of negation and certainty scope resolution is quite high. Since we have event annotations, we might post-process the output of these models to see if it scopes over the event.
- **Active Learning:** Because SemRep and SemRep Factuality perform poorly on only a few labels, active learning techniques could be leveraged to obtain annotations only for these labels.

In Chapter 1 we briefly mentioned that while polarity and uncertainty are fundamental dimensions of factuality, each and every context dimension has its own bearing on factuality. Essentially, a context dimension can either modify the epistemic uncertainty of an event by adding restrictions (e.g., an intervention is effective for laboratory mice but unknown for humans) or it can indicate additional aleatoric uncertainty (e.g., a study conducted on a small sample). Ideally, the author of a event would take all relevant context into account when explicitly expressing the factuality of an event, but we cannot rely upon this. Therefore, to completely determine the factuality of an event, extensive future work is required to 1) identify various context dimensions in text, 2) resolve them to their target events, which may

lie sentences or paragraphs away, and 3) map the values of the context dimensions to values of polarity and uncertainty. Once this mapping has been done, the polarity and uncertainty values can be integrated into the belief representations introduced in Section 3.4.1 using existing SL operators.

# Chapter 4

# Disentangling Event Content from Polarity and Uncertainty

As already described, the factuality of an event is composed of its polarity (expressed using negation) and uncertainty (expressed using speculation). In formal semantics, negation and speculation are operators whose semantic functions are independent of the propositional content they modify [183], [184]. More formally, the propositional content can be replaced by a variable such that it is possible to form fluent statements by varying only one of these aspects while leaving the others the same[1]. Polarity, uncertainty, and propositional content can thus be viewed as *independent generative factors* of events. An example of this is given in Figure 4.1.

Figure 4.1. Example indicating the distinction between polarity, uncertainty, and content. The content "Trees have leaves" is modified by the negation (**bold**) and speculation (underlined) expressions.

While this is simple enough to show, it remains an open question as to whether the internal representations of these generative factors learned by neural network models can reflect this independence. The task of learning independent representations of generative factors is Disentangled Representation Learning (DRL). DRL can improve the robustness of representations and their applicability across tasks and is an important ingredient for interpretable models [185]. Previous works have disentangled attributes such as syntax, semantics, and style [186]–[191], but the disentanglement of polarity, uncertainty, and propositional content has not yet been investigated.

The first part of this chapter, adapted from research previously published by the author in the Proceedings of the Association for Computational Linguistics, explores the disentanglement of polarity and uncertainty on general-domain text using a deep generative model called a Variational AutoEncoder (VAE) [192]. Specifically, it answers the following research questions:

**RQ1:** Is it possible to estimate a model of statements that upholds the proposed statistical independence between polarity, uncertainty, and content?

[1] E.g., given two negated propositions $\neg p$ and $\neg r$, the function of the negation operator $\neg$ does not change whether it is attached to $p$ or $r$. Likewise, the meanings of $p$ and $r$ do not change *per se* when they are negated.

**RQ2:** A number of existing disentanglement objectives have been explored for text, all giving promising results. How do these objectives compare for enforcing disentanglement on this task?

Given the results of our experiments and answers to the above research questions, part 2 (Section 4.7 onwards) applies our findings to develop two updated models, SubjVAE and FactVAE. These models are transformer-based VAEs that leverage Subjective Logic (SL) distributions for modeling the latent variables. We show that both models successfully disentangle polarity, uncertainty, and predicate type on SemMed Context. Additionally, FactVAE learns a single SL Beta distribution that is predictive of both polarity and uncertainty, meaning it can be used directly in downstream knowledge fusion applications, such as the knowledge graph constructed in Section 3.4.

## 4.1 Background

We here provide the necessary background on VAEs and disentangled representation learning to understand the remainder of this chapter. We refer the interested reader to the more in-depth discussions of these topics in [193] (VAEs), and [185] (DRL).

### 4.1.1 Variational AutoEncoders

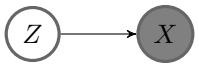



Figure 4.2. Graph of the basic generative model estimated by a VAE. $X$ is the observed data and $Z$ is the unobserved latent.

The goal of the VAE is to estimate a probabilistic generative model of the observed input variable $X$ given a latent unobserved variable $Z$, depicted in Figure 4.2 [194]. That is, we want to estimate the parameters $\theta$ of the following marginal probability density.

$$p(x) = \int_{z \in Z} p_\theta(X|z)p(z)dz$$

where $p_\theta(X|Z)$ is the likelihood of the data and $p(Z)$ is a prior distribution over $z$. To estimate $\theta$ having observed only $X$, we need to estimate the posterior distribution $p(Z|X)$, which can be defined as

$$p(Z|X) = \frac{p_\theta(X|Z)p(Z)}{p(X)}$$

However, except in special cases, this does not generally have a closed-form solution. Variational Inference (VI) circumvents this by computing an approximation of $p(Z|X)$, denoted $q_\phi(Z|X)$, by maximizing the Evidence Lower BOund (ELBO).

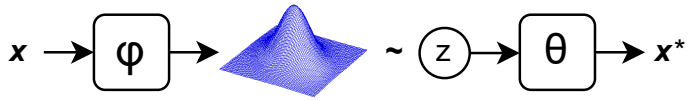



Figure 4.3. A diagram of a Variational AutoEncoder. The input data $x$ is encoded by a neural network $\phi$ into a latent distribution. The distribution is sampled to obtain a latent value $z$, which is decoded by another neural network $\theta$ that attempts to reconstruct the input.

$$\begin{aligned}
\text{ELBO} &= \mathbb{E}_{q_\phi(Z|X)} \log \left[ \frac{p(X)}{q_\phi(Z|X)} \right] \\
&= \mathbb{E}_{q_\phi(Z|X)} \log \left[ \frac{p(X|Z)p(Z)}{q_\phi(Z|X)} \right] \\
&= \mathbb{E}_{q_\phi(Z|X)} \big[ \log p(X|Z) \big] - KL \big[ q_\phi(Z|X) || p(Z) \big]
\end{aligned}$$

where $KL[q||p]$ is the Kullback-Leibler divergence between the probability distributions $p$ and $q$. Intuitively, the ELBO aims to maximize the log probability of the data under the approximated posterior regularized to closely match the prior. In a VAE, both the approximate posterior $q_\phi(Z|X)$ and the likelihood $p_\theta(X|Z)$ are parameterized using neural networks. The $q$ network is the encoder, which transforms the input into the parameters $\phi$ of the latent probability distribution. This distribution is sampled to obtain a latent value $z$ which is passed to the decoder network $p$, which attempts to reconstruct the input. A diagram of this process is given in Figure 4.3.

Because VAEs are estimated using back-propagation, it is necessary to compute the gradients of the samples from the latent distributions. However, standard sampling approaches are non-differentiable due to the inherent randomness of the process. This is circumvented using the reparameterization trick for Gaussian distributions [194]. Given the $\mu, \sigma$ parameters of a Gaussian predicted by the encoder, we can compute a differentiable sample as

$$z = \mu + \sigma \cdot \epsilon$$

where $\epsilon \sim \mathcal{N}(0, 1)$. The reparameterization trick essentially relocates the source of randomness from the parameters to $\epsilon$, which allows the gradient of $\mu$ and $\sigma$ to be calculated.

### 4.1.2 Disentangled Representation Learning

There is currently no agreed-upon definition of disentanglement. Early works on DRL attempt to learn a single vector space in which each dimension is independent of the others and represents one ground-truth generative factor of the object being modeled [195]. [196] give a group-theoretic definition, according to which generative factors are mapped to independent vector spaces. This definition relaxes the earlier assumption that representations ought to be single-dimensional and formalizes the notion of disentanglement according to the notion of invariance. [197] decompose the invariance requirement into consistency and restrictiveness,

which describe specific ideal properties of the invariances between representations and generative factors. In addition to independence and invariance, interpretability is an important criterion for disentanglement. [195] point out that while methods such as PCA are able to learn *independent* latent representations, because these are not representative of interpretable factors of variation, they are not *disentangled*. We therefore want our learned representations to be predictive of meaningful factors of variation. We adopt the term *informativeness* from [198] to signify this desideratum.

Previous works on DRL for text all use some form of supervision to enforce informativeness of the latent representations. [190], [187], [188], and [189] all use gold-standard labels of the generative factors, while other works employ similarity metrics [186], [199]. In contrast, our approach uses weak labels for polarity and uncertainty generated using a classifier trained on a small set of gold-standard data.

These previous works on text DRL all use a similar architecture: a sequence VAE [194], [200] maps text inputs to $L$ distinct latent spaces, each of which are constrained to represent a different target generative factor via a supervision signal. We also employ this overall architecture for model estimation and use it as a basis for experimenting with existing disentanglement objectives based on adversarial learning [187], [189] and mutual information minimization [188], described in Section 4.3.2. However, unlike these previous works, which learn high-dimensional representations of all the latent factors, we aim to learn 1-dimensional representations of the polarity and uncertainty variables.

### 4.1.3 Contributions

In addressing these research questions, we make the following contributions:

1. *Generative Model:* We propose a generative model of statements in which polarity, uncertainty, and event content are independent latent variables. Following previous works, we estimate this model using a VAE and compare existing auxiliary objectives for enforcing disentanglement via a suite of evaluation metrics.
2. *Simple Latent Representations:* We presume polarity and uncertainty to be binary (positive or negative, certain or uncertain)[2]. We therefore attempt to learn corresponding 1-dimensional latent representations for these variables, with a clear separation between each value.
3. *Data Augmentation:* Datasets containing polarity and uncertainty annotations are relatively small [201]–[203], resulting in poor sentence reconstructions according to our preliminary experiments. To address this, we generate weak labels for a large number of Amazon[3] and Yelp[4] reviews using a simple naïve Bayes classifier with bag-of-words features trained on a smaller dataset of English reviews annotated for polarity and uncertainty [178] and use this to estimate our model. Details are given in Section 4.4.1.

[2]This is a simplification for uncertainty, but one necessitated by the labels available.
[3]`https://github.com/fuzhenxin/text_style_transfer`
[4]`https://github.com/shentianxiao/language-style-transfer4`

We note that, in contrast to other works on polarity and uncertainty modeling, which focus on token-level tasks of negation and speculation cue and scope detection, this work aims to learn *statement*-level representations of our target factors, in line with previous work on text DRL. We describe our overall model in Section 4.2. Section 4.3 enumerates three specific desiderata for disentangled representations, and sections 4.3.1 and 4.3.2 describe how we aim to satisfy these desiderata.

## 4.2 Generative Model

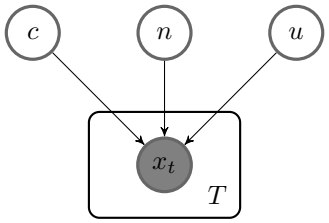



Figure 4.4. Graph of the generative model. $c$ is content, $n$ is polarity (negation status), and $u$ is uncertainty.

We propose a generative model of statements according to which polarity, uncertainty, and content are independent latent variables. A diagram of our proposed model is given in Figure 4.4. We here define our generative model and derive the corresponding ELBO objective for our proposed VAE with three latent variables. Let $x = [x_1, ..., x_T]$ be a sentence with $T$ tokens. $P$, $U$, and $C$ are the latent variables representing polarity (negation status), uncertainty, and content respectively. The joint probability of specific values of these variables ($N = n$, $U = u$, $C = c$) is defined as

$$p(x, n, u, c) = p_\theta(x|n, u, c)p(n)p(u)p(c) \tag{4.1}$$

Furthermore, $x$ is defined auto-regressively as

$$p(x|n, u, c) = \prod_t p(x_t|x_{<t}, n, u, c) \tag{4.2}$$

Our model assumes that the latent factors are independent, so the posterior is

$$p(n, u, c|x) = p(n|x)p(u|x)p(c|x) \tag{4.3}$$

$$= \prod_{\ell \in \{n,u,c\}} \frac{p(x|\ell)p(\ell)}{p(x)} \tag{4.4}$$

We approximate the posterior $p(n, u, c|x)$ with $q_\phi(n, u, c|x)$. Because the posterior factors, we approximate the individual posteriors $p(\cdot|x)$ with $q_\phi(\cdot|x)$ and derive the following ELBO.

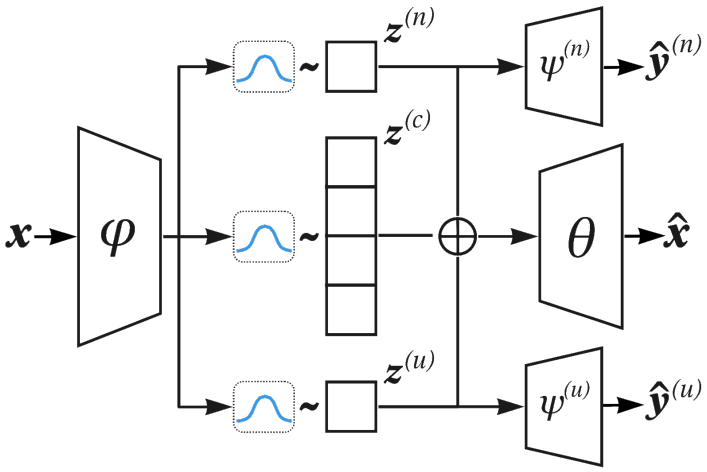



Figure 4.5. The proposed architecture corresponding to the $\mathcal{L}_{ELBO} + \mathcal{L}_{INF}$ objective (see Section 4.3.2). A BiLSTM encoder parameterized by $\phi$ maps each input example $x$ to three distinct distributions from which the latent representations $z^{(\ell)}$ are sampled. The negation $z^{(n)}$ and uncertainty $z^{(u)}$ latent spaces are then passed to linear classifiers, parameterized by $\psi^{(\ell)}$, which attempt to predict the ground truth factor. Finally, the latent values initialize an LSTM decoder, parameterized by $\theta$, which attempts to reconstruct the input.

$$\text{ELBO} = \mathbb{E}_{q_\phi(n,u,c|x)}\left[\log \frac{p(x,n,u,c)}{q_\phi(n,u,c|x)}\right] \tag{4.5}$$

$$= \mathbb{E}_{q_\phi}\left[\log \frac{p(x|n,u,c)p(n)p(u)p(c)}{q_\phi(n|x)q_\phi(u|x)q_\phi(c|x)}\right] \tag{4.6}$$

$$= \mathbb{E}_{q_\phi}\left[\log p(x|n,u,c)\right] - \sum_{\ell} KL\left[q_\phi(\ell|x)||p(\ell)\right] \tag{4.7}$$

$$\tag{4.8}$$

In our implementation, three linear layers map the final hidden state of a BiLSTM encoder to three sets of Gaussian distribution parameters ($\mu$, $\sigma$), which parameterize the polarity, uncertainty, and content latent distributions $\ell \in \{n, u, c\}$, respectively. Because we map each input to three distinct latent spaces, we include three KL divergence terms in the ELBO training objective, given in Equation (4.9).

$$\mathcal{L}_{\text{ELBO}}(\phi,\theta) = -\mathbb{E}_{q_\phi(z|x)}\Big[\log p_\theta(x|z)\Big] + \sum_{\ell\in\{n,u,c\}} \beta_\ell \,\text{KL}\Big[q_\phi^{(\ell)}(z^{(\ell)}|x) \;||\; p(z^{(\ell)})\Big] \tag{4.9}$$

where $\phi$ denotes the encoder's parameters, $\theta$ the decoder's parameters, $p(z^{(\ell)})$ is a standard Gaussian prior, and the $\beta_\ell$ hyper-parameters weight the KL divergence term for each latent space $\ell \in L$. The latent representations $z^\ell$ are sampled from normal distributions defined by these parameters using the reparameterization trick. The latent representations are then concatenated $z = [z^{(n)}; z^{(u)}; z^{(c)}]$ and used to initialize an LSTM decoder, which aims to reconstruct the input. A visualization of our architecture is given in Figure 4.5 and implementation details are given in Section 4.5.

We use 1-dimensional polarity and uncertainty spaces and a 62-dimensional content space for a total latent size of 64. Notably, we do not supervise the content space, unlike previous works [187], [188], which supervise it by predicting the bag of words of the input. Such a supervision technique would hinder disentanglement by encouraging the content space to be predictive of the polarity and uncertainty cues. Therefore, in our model we define three latent spaces $\ell \in \{n, u, c\}$ but use signals from only 2 target generative factors $k \in \{n, u\}$.

## 4.3 Desiderata for Disentanglement

We aim to satisfy the following desiderata of disentangled representations put forth by previous works.

1. *Informativeness*: the representations should be predictive of the ground-truth generative factors [195], [198];
2. *Independence*: the representations for each generative factor in question should lie in independent vector spaces [196];
3. *Invariance*: the mapping from the data to the representations should be invariant to changes in other generative factors [196], [197];

The following sections detail how our model enforces these desiderata.

### 4.3.1 Informativeness

Following [198], we measure the informativeness of a representation by its ability to predict the corresponding generative factor. Similar to previous works on DRL for text [187], [188], we train supervised linear classifiers[5] on each latent space and back-propagate the prediction error. Thus, in addition to the ELBO objective in Equation (4.9), we define informativeness objectives for polarity and uncertainty.

$$\mathcal{L}_{\text{INF}}(\psi^{(k)}) = \text{BCE}\Big(\hat{y}^{(k)}, y^{(k)}\Big), \;\; k \in \{n, u\} \tag{4.10}$$

where $y^{(k)}$ is the true label for factor $k$, $\hat{y}^{(k)}$ is the classifier's prediction, $\psi^{(k)}$ are the parameters of this classifier, and BCE is the binary cross-entropy loss.

### 4.3.2 Independence and Invariance

We compare 3 objectives for enforcing these desiderata:

[5] Implemented as single-layer feed-forward neural networks with sigmoid activation.

1. Informativeness (INF): This is based on the hypothesis that if polarity, uncertainty, and content are independent generative factors, the informativeness objective described in Section 4.3.1 will be sufficient to drive independence and invariance. This approach was found to yield good results on disentangling style from content in [186].
2. Adversarial (ADV): The latent representations should be predictive of their target generative factor only. Therefore, inspired by [187], we train additional adversarial classifiers on each latent space that try to predict the values of the non-target generative factors, while the model attempts to structure the latent spaces such that the predictive distribution of these classifiers is a non-predictive as possible.
3. Mutual-information minimization (MIN): A natural measure of independence between two variables is mutual information (MI). Therefore, this objective minimizes an upper-bound estimate of the MI between each pair of latent spaces, following [188], [191], [204].

Details on the ADV and MIN objectives are given below.

**Adversarial Objective:** The adversarial objective (ADV) consists of two parts: 1) adversarial classifiers which attempt to predict the value of all non-target factors from each latent space; 2) a loss that aims to maximize the entropy of the predicted distribution of the adversarial classifiers.

For a given latent space $\ell$, a set of linear classifiers predict the value of each non-target factor $k \neq \ell$, respectively, and we compute the binary cross-entropy loss for each.

$$\mathcal{L}_{\text{CLS}}(\xi^{(\ell,k)}) = \text{BCE}\left(\hat{y}^{(\ell,k)}, y^{(k)}\right) \tag{4.11}$$

Where $\xi^{(\ell,k)}$ are the parameters of the adversarial classifier predicting factor $k$ from latent space $\ell$, and $\hat{y}^{(\ell,k)}$ is the corresponding prediction.

For example, we introduce two such classifiers for the content space $\ell = c$, one to predict polarity and one to predict uncertainty, $k \in \{n, u\}$. Importantly, the prediction errors of these classifiers are not back-propagated to the rest of the VAE. We impose an additional objective for each adversarial classifier, which aims to make it's predicted distribution as close to uniform as possible. We do this by maximizing the entropy of the predicted distribution (Equation (4.12)) and back-propagating the error, following [187] and [205].

$$\mathcal{L}_{\text{ENT}}(\xi^{(\ell,k)}) = H[\hat{y}^{(\ell,k)}] \tag{4.12}$$

As the objective is to maximize this quantity, the total adversarial objective is

$$\mathcal{L}_{\text{ADV}} = \sum_{\ell} \sum_{k} \mathcal{L}_{\text{CLS}}(\xi^{(\ell,k)}) - \mathcal{L}_{\text{ENT}}(\xi^{(\ell,k)}) \tag{4.13}$$

The ADV objective aims to make the latent representations as uninformative as possible for non-target factors. Together with the informativeness objective, it pushes the representations to specialize in their target generative factors.

**MI Minimization Objective:** The MI minimization (MIN) objective focuses on making the distributions of each latent space as dissimilar as possible. We minimize the MI between each pair of latent spaces according to Equation (4.14).

$$\mathcal{L}_{\text{MIN}} = \hat{I}_{\text{CLUB}}(\ell_i; \ell_j), \;\; i \neq j \tag{4.14}$$

where $\hat{I}_{\text{CLUB}}(\ell_i; \ell_j)$ is the Contrastive Learning Upper-Bound (CLUB) estimate of the MI [204]. Specifically, we introduce a separate neural network to approximate the conditional variational distribution $p_\sigma(\ell_i|\ell_j)$, which is used to estimate an upper bound on the MI using samples from the latent spaces.

The full model objective along with relevant hyperparameters weights $\boldsymbol{\lambda}$ is given in Equation (4.15). Our hyperparameter settings and further implementation details are given in Section 4.5.

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \lambda_{\text{INF}}\mathcal{L}_{\text{INF}} + \lambda_{\text{ADV}}\mathcal{L}_{\text{ADV}} + \lambda_{\text{MIN}}\mathcal{L}_{\text{MIN}} \tag{4.15}$$

In the sections that follow, we experiment with different subsets of the terms in the full objective and their effects on disentanglement. We train a model using only the ELBO objective as our disentanglement baseline.

## 4.4 Experiments

We describe our datasets, preprocessing, and data augmentation methods in Section 4.4.1. Section 4.4.2 describes our evaluation metrics and how these target the desiderata for disentanglement given in Section 4.3.

### 4.4.1 Datasets

We use the SFU Review Corpus [178] as our primary dataset. This corpus contains 17,000 sentences from reviews of various products in English, originally intended for sentiment analysis, annotated with negation and speculation cues and their scopes. Many of the SFU sentences are quite long ($> 30$ tokens), and preliminary experiments revealed that this results

in poor reconstructions. We therefore took advantage of SFU's annotated statement conjunction tokens to split the multi-statement sentences into single-statement ones in order to reduce the complexity and increase the number of examples. Also to reduce complexity, we remove sentences $> 15$ tokens following previous work [190], resulting in 14,000 sentences.

We convert all cue-scope annotations to statement-level annotations. Multi-level uncertainty annotations have been shown to be rather inconsistent and noisy, achieving low inter-annotator agreement compared to binary ones [206]. We therefore binarize the certainty labels following [133].

#### Data Augmentation

Despite the efforts above, we found the SFU corpus alone was insufficient for obtaining fluent reconstructions. We therefore generated weak polarity and uncertainty labels for a large number of additional Amazon and Yelp review data using two naïve Bayes classifiers with bag-of-words (BOW) features. These classifiers were trained on the SFU training split to predict sentence level polarity and uncertainty, respectively. We include these weakly-labeled examples in the training, validation, and test splits of our datasets. While it is perhaps not ideal to evaluate our models on weakly-labeled examples due to noise coming from the naïve Bayes classifiers, we justify this choice in two ways: first, the evaluation of the naïve Bayes classifiers in Table 4.2 shows good performance and thus relatively low amounts of label noise; second, the increase in total number of evaluation examples as a result of the weak labels reduces the overall evaluation uncertainty compared to an evaluation on the SFU examples only.

We performed feature selection by computing the $K$ tokens from the SFU training data that had the highest ANOVA F-value against the target labels, implemented using `f_classif` in scikit-learn [207]. We tuned $K$ according to the downstream classification performance on the SFU dev set. We evaluated $K$ in the range 3-30 and found $K = 20$ performed best for both models. The 20 tokens ultimately used as features by the negation and speculation classifiers are given in Table 4.1.

| | |
|---|---|
| Negation | *any, but, ca, cannot, did, do, does, dont, either, even, have, i, it, n't, need, never, no, not, without, wo* |
| Speculation | *be, can, could, either, i, if, may, maybe, might, must, or, perhaps, probably, seem, seemed, seems, should, think, would, you* |

Table 4.1. The tokens used as features for the negation and speculation classifiers. Contractions are split in our tokenization scheme.

The precision, recall, and F1 scores of both classifiers on the SFU dev and test sets are given in Table 4.2.

The Amazon and Yelp datasets fit the SFU data distribution well, being also comprised of user reviews, and have been used in previous works on text DRL with good results [187],

| | | P | R | F1 |
|---|---|---|---|---|
| Polarity | dev | 0.942 | 0.877 | 0.901 |
| | test | 0.946 | 0.880 | 0.909 |
| Uncertainty | dev | 0.959 | 0.953 | 0.956 |
| | test | 0.948 | 0.961 | 0.954 |

Table 4.2. Precision, recall, and F1 score of the negation and uncertainty classifiers on the SFU dev and test sets.

[188][6]. Statistics for the combined SFU+Amazon dataset are summarized in Table 4.3.

| | **SFU+Amazon** | | | **SFU+Yelp** | | |
|---|---|---|---|---|---|---|
| | N | % Negative | % Uncertain | N | % Negative | % Uncertain |
| Train | 109,889 | 22.7% | 17.9% | 390,409 | 14.5% | 8.8% |
| Dev | 6,631 | 19.6% | 15.2% | 5,971 | 15.5% | 9.1% |
| Test | 6,579 | 20.4% | 15.2% | 3,118 | 12.9% | 10.4% |

Table 4.3. Statistics for the combined SFU and weakly labeled Amazon and Yelp datasets.

### 4.4.2 Evaluation

Evaluating disentanglement of the learned representations requires complementary metrics of the desiderata given in Section 4.3: informativeness, independence, and invariance.

For measuring informativeness, we report the precision, recall, and F1 score of a logistic regression model trained to predict each of the ground-truth labels from each latent space, following [198]. We also report the MI between each latent distribution and factor, as this gives additional insight into informativeness.

For measuring independence, we use the Mutual Information Gap (MIG) [208]. The MIG is the difference in MI between the top-2 latent variables $\ell_{\{i,j\}}$ with respect to a given generative factor $k$, normalized to lie in $[0, 1]$, with higher values indicating a greater degree of disentanglement.

$$\text{MIG}_k = \frac{1}{H[k]} \left( I(\mathbf{z}^{(\ell_i)}; k) - \max_{j \neq i} \; I(\mathbf{z}^{(\ell_j)}; k) \right) \tag{4.16}$$

where $\ell_i = \text{argmax}_{i'} \; I(\mathbf{z}^{(\ell_{i'})}; k)$ is the latent space with the highest MI with the generative factor. We estimate the MI between latent representations and labels using the method proposed by [209], implemented using `mutual_info_classif` in scikit-learn [207] using 30 resamples from the predicted latent distributions for each example.

We evaluate invariance by computing the Pearson's correlation coefficient between each pair of latent variables using samples from the predicted latent distributions.

It also important to evaluate the ability of the models to reconstruct the input. Specifically, we target reconstruction faithfulness (i.e., how well the input and reconstruction match) and

[6]Due to computational constraints, we randomly sample 100,000 weakly annotated Amazon examples for the final dataset. Preliminary experiments with larger numbers of Amazon examples suggested that 100k is sufficient for our purposes.

fluency. We evaluate faithfulness in terms of the ability of the model to preserve the polarity, uncertainty, and content of the input. Polarity and uncertainty preservation are measured by re-encoding the reconstructions, predicting polarity and uncertainty from the re-encoded latent values, and computing precision, recall, and F1 score against the ground-truth labels[7]. Following previous work, we approximate a measure of content preservation in the absence of any explicit content annotations by computing the BLEU score between the input and the reconstruction (self-BLEU) [186], [188], [189]. We evaluate fluency of the reconstruction by computing the perplexities (PPL) under GPT-2, a strong, general-domain language model [108].

Finally, we evaluate the models' ability to flip the polarity or uncertainty status of the input. For each test example, we override the value of the latent factor we want to change to represent the *opposite* of its ground-truth label. The ability of the model to control polarity and uncertainty is measured by re-encoding the reconstructions obtained from the overridden latents, predicting from the re-encoded latent values, and computing accuracy against the *opposite* of the ground-truth labels.

## 4.5 Implementation Details

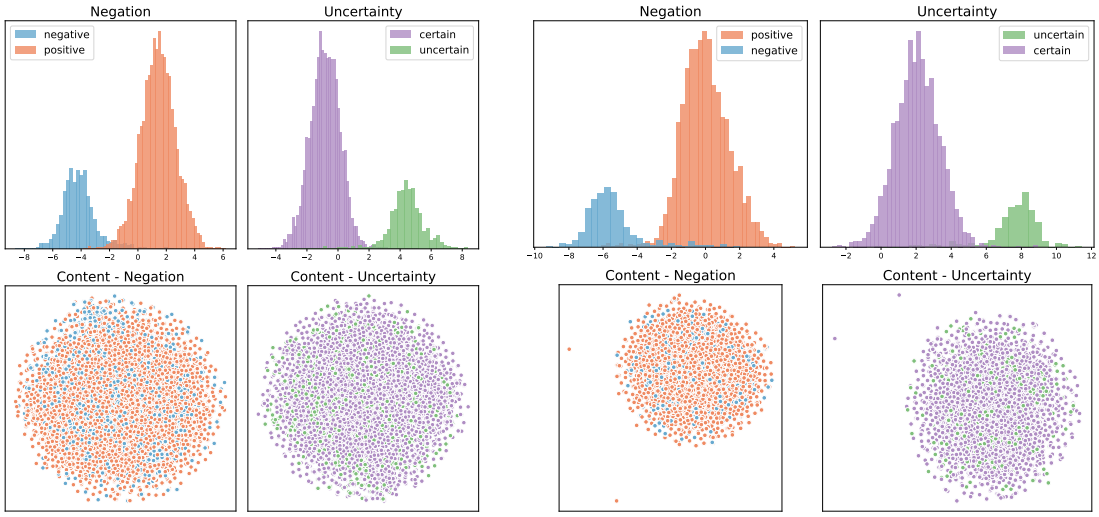
We implement our model in PyTorch [210]. The encoder is a BiLSTM with 2 hidden layers and hidden size 256. The decoder is a 2 layer LSTM with hidden size 256. Embeddings for both the encoder and decoder are of size 256 and are randomly initialized and learned during training. The encoder and decoder use both word dropout and hidden layer dropout between LSTM layers, with a dropout probability of 0.5. We also use teacher forcing when training the decoder, with a probability of using the true previous token set to 0.5. We trained each model for 20 epochs with a batch size of 128 and the ADAM optimizer [211] with a learning rate of $3 \cdot 10^{-4}$. Training took around 6.5 hours for each model on one Tesla v100 with 16GB of VRAM.

The latent space classifiers for the INF objective and the adversarial classifiers for the ADV objective both used a single linear layer with sigmoid activation. The adversarial classifiers were trained with a separate ADAM optimizer with learning rate of $3 \cdot 10^{-4}$. For MI estimation as part of the MIN objective we used the code released alongside the CLUB paper [204] [8]. For training the approximation network, we again use an ADAM optimizer with learning rate $5 \cdot 10^{-4}$.

All hyperparameter weights were tuned by hand. We weight the KL divergence term of each latent space separately as follows: $\beta_c = 0.01,\ \beta_n = \beta_u = 0.005$. We experimented with both higher KL weights and KL annealing schedules, but found that they did not improve disentanglement and higher weights tended to negatively impact reconstruction ability. For the individual objectives, we set the following weights: $\lambda_{INF} = 1.0,\ \lambda_{ADV} = 1.0,\ \lambda_{MIN} = 0.01$. While we found the model relatively robust to different values of $\lambda_{INF}$ and $\lambda_{ADV}$, the

[7] This corresponds to the measure of consistency proposed by [197]

[8] `https://github.com/Linear95/CLUB`




(a) SFU+Amazon (b) SFU+Yelp

Figure 4.6. Histogram and t-SNE visualizations of the polarity (negation), uncertainty, and content spaces learned by the INF+ADV+MIN model.

MIN objective was found to be quite sensitive to even small changes of both $\lambda_{MIN}$ and the learning rate of the MI approximation network.

## 4.6 Results

This section reports the results on latent disentanglement as well as the faithfulness and fluency of the models' reconstructions. We also discuss how these results address the two research questions posed at the beginning of the chapter.

### 4.6.1 Disentanglement

The informativeness of each latent space with respect to each target factor is shown in Table 4.4 given as predictive performance and MI.

**SFU+Amazon**

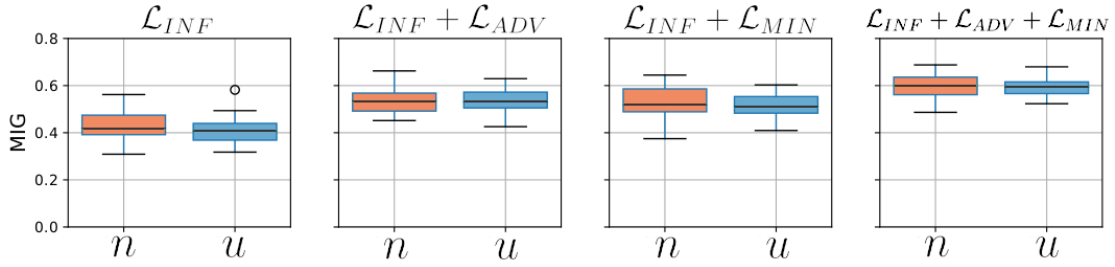
| | | $\mathcal{L}_{\text{ELBO}}$ | | | | +$\mathcal{L}_{\text{INF}}$ | | | | +$\mathcal{L}_{\text{INF}}$+$\mathcal{L}_{\text{ADV}}$ | | | | +$\mathcal{L}_{\text{INF}}$+$\mathcal{L}_{\text{MIN}}$ | | | | +$\mathcal{L}_{\text{INF}}$+$\mathcal{L}_{\text{ADV}}$+$\mathcal{L}_{\text{MIN}}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Latent | Factor | MI | P | R | F1 | MI | P | R | F1 | MI | P | R | F1 | MI | P | R | F1 | MI | P | R | F1 |
| $n$ | $n$ | 0.02 | 0.56 | 0.62 | 0.53 | 0.43 | 0.95 | 0.97 | 0.96 | 0.43 | 0.95 | 0.97 | 0.96 | 0.44 | 0.96 | 0.98 | 0.97 | 0.44 | 0.96 | 0.98 | 0.97 |
| | $u$ | 0.02 | 0.56 | 0.59 | 0.53 | 0.01 | 0.55 | 0.58 | 0.54 | 0.01 | 0.55 | 0.58 | 0.55 | 0.01 | 0.54 | 0.56 | 0.53 | 0.01 | 0.54 | 0.57 | 0.54 |
| $u$ | $n$ | 0.01 | 0.56 | 0.61 | 0.54 | 0.01 | 0.56 | 0.58 | 0.55 | 0.01 | 0.55 | 0.57 | 0.55 | 0.01 | 0.54 | 0.56 | 0.54 | 0.01 | 0.55 | 0.56 | 0.55 |
| | $u$ | 0.02 | 0.57 | 0.60 | 0.56 | 0.38 | 0.94 | 0.97 | 0.95 | 0.37 | 0.94 | 0.97 | 0.96 | 0.39 | 0.96 | 0.98 | 0.97 | 0.39 | 0.97 | 0.98 | 0.98 |
| $c$ | $n$ | 0.30 | 0.68 | 0.76 | 0.70 | 0.22 | 0.68 | 0.75 | 0.68 | 0.17 | 0.58 | 0.62 | 0.55 | 0.18 | 0.64 | 0.71 | 0.64 | 0.15 | 0.58 | 0.62 | 0.56 |
| | $u$ | 0.21 | 0.65 | 0.76 | 0.67 | 0.20 | 0.64 | 0.75 | 0.65 | 0.14 | 0.57 | 0.63 | 0.54 | 0.17 | 0.64 | 0.74 | 0.64 | 0.14 | 0.57 | 0.64 | 0.55 |

**SFU+Yelp**

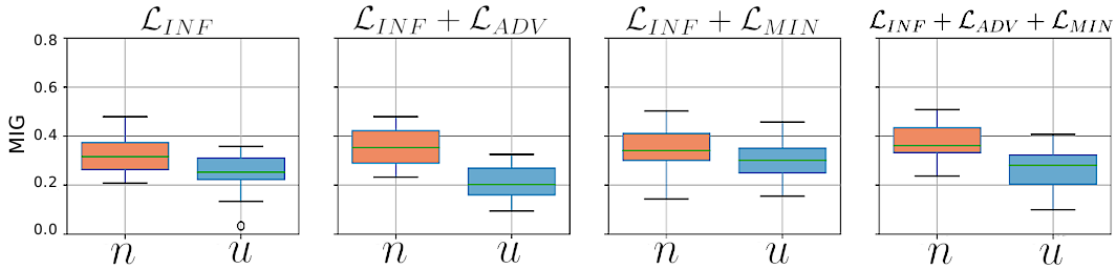
| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $n$ | 0.02 | 0.56 | 0.63 | 0.56 | 0.31 | 0.92 | 0.95 | 0.93 | 0.31 | 0.91 | 0.95 | 0.93 | 0.31 | 0.91 | 0.95 | 0.93 | 0.31 | 0.92 | 0.95 | 0.94 |
| | $u$ | 0.01 | 0.53 | 0.56 | 0.51 | 0.01 | 0.54 | 0.60 | 0.53 | 0.01 | 0.56 | 0.63 | 0.56 | 0.01 | 0.55 | 0.61 | 0.54 | 0.01 | 0.56 | 0.62 | 0.55 |
| $u$ | $n$ | 0.02 | 0.55 | 0.63 | 0.51 | 0.01 | 0.54 | 0.58 | 0.53 | 0.02 | 0.54 | 0.58 | 0.53 | 0.01 | 0.54 | 0.58 | 0.53 | 0.02 | 0.54 | 0.58 | 0.53 |
| | $u$ | 0.04 | 0.58 | 0.67 | 0.56 | 0.28 | 0.93 | 0.96 | 0.94 | 0.28 | 0.92 | 0.96 | 0.94 | 0.28 | 0.92 | 0.96 | 0.94 | 0.28 | 0.92 | 0.96 | 0.94 |
| $c$ | $n$ | 0.24 | 0.65 | 0.78 | 0.67 | 0.18 | 0.60 | 0.71 | 0.59 | 0.17 | 0.58 | 0.66 | 0.55 | 0.17 | 0.58 | 0.66 | 0.55 | 0.16 | 0.56 | 0.63 | 0.53 |
| | $u$ | 0.23 | 0.62 | 0.76 | 0.62 | 0.20 | 0.59 | 0.72 | 0.58 | 0.21 | 0.58 | 0.70 | 0.57 | 0.18 | 0.58 | 0.69 | 0.56 | 0.20 | 0.58 | 0.68 | 0.56 |

Table 4.4. Informativeness results on SFU+Amazon and SFU+Yelp showing the Mutual Information estimate between each latent and factor as well as Precision, Recall, and F1 of the latent space classifiers for each factor. $n$: polarity (negation). $u$: uncertainty. $c$: content. Shown are the mean values computed from 30 resamples of the latent distributions for each example.

On both datasets the baseline ELBO objective alone fails to disentangle. It puts almost all representation power in the content space, which is nevertheless still uninformative of the polarity and uncertainty factors, with low F1s. The model using the INF auxiliary objective does, however, manage to achieve good disentanglement: the polarity and uncertainty spaces are highly informative of their target factors and uninformative of the non-target factors[9]. However, the content space is still slightly predictive of polarity and uncertainty on SFU+Amazon, with F1s over 0.6 for both tasks, although this does not occur on SFU+Yelp. The ADV and MIN objectives improve disentanglement on SFU+Amazon as the content space shows near-random prediction performance of polarity and uncertainty, with slightly improved prediction performance of the polarity and uncertainty spaces for their target factors. These results are corroborated by the visualizations in Figure 4.6, which show clear separation by classes in the polarity and uncertainty latent distributions but no distinction between classes in the content space. Additionally, we note the good predictive performance of the polarity and uncertainty latents, despite their simple, 1-dimensional encoding.




(a) SFU+Amazon




(b) SFU+Yelp

Figure 4.7. Box plots of the Mutual Information Gap (MIG) for each disentanglement objective for the negation and uncertainty factors computed on the test sets. The MIG values for the baseline $\mathcal{L}_{\text{ELBO}}$ objective were too small to include in this figure.

Box plots of the MIG for each disentanglement objective are given in Figure 4.7. In general, there is less disentanglement on SFU+Yelp than on SFU+Amazon (MIG $\approx 0.4$ vs MIG $\approx 0.6$ on SFU+Amazon) and while the auxiliary disentanglement objectives improve MIG on SFU+Amazon, their effect is less pronounced on SFU+Yelp. A comparison of the MI values reported in Table 4.4 shows a possible cause: the polarity and uncertainty latents on SFU+Yelp are less predictive of their respective target factors (MIs $\approx 0.3$ vs $\approx 0.4$ on SFU+Amazon) while the content space is more predictive of these factors (e.g., content-uncertainty MI = 0.196 on SFU+Yelp vs 0.136 on SFU+Amazon). This may be due to a data mismatch between SFU and Yelp, since SFU is a dataset of product reviews, while Yelp contains mostly restaurant and store reviews.

[9]Experiments using $\mathcal{L}_{\text{ELBO}}$+$\mathcal{L}_{\text{ADV}}$ or $\mathcal{L}_{\text{ELBO}}$+$\mathcal{L}_{\text{MIN}}$ did not improve over $\mathcal{L}_{\text{ELBO}}$ alone.

**SFU+Amazon**

| | $\mathcal{L}_{\text{ELBO}}$ | | $+\mathcal{L}_{\text{INF}}$ | | $+\mathcal{L}_{\text{INF}}$ $+\mathcal{L}_{\text{ADV}}$ | | $+\mathcal{L}_{\text{INF}}$ $+\mathcal{L}_{\text{MIN}}$ | | $+\mathcal{L}_{\text{INF}}+\mathcal{L}_{\text{ADV}}$ $+\mathcal{L}_{\text{MIN}}$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $u$ | $c$ | $u$ | $c$ | $u$ | $c$ | $u$ | $c$ | $u$ | $c$ |
| $n$ | 0.706 | 0.008 ($\pm$0.053) | 0.200 | 0.002 ($\pm$0.098) | 0.193 | 0.007 ($\pm$0.076) | 0.129 | 0.008 ($\pm$0.058) | 0.159 | 0.001 ($\pm$0.043) |
| $u$ | - | 0.001 ($\pm$0.058) | - | 0.001 ($\pm$0.097) | - | 0.010 ($\pm$0.070) | - | 0.001 ($\pm$0.060) | - | 0.005 ($\pm$0.037) |

**SFU+Yelp**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | 0.090 | 0.003 ($\pm$0.013) | 0.135 | 0.001 ($\pm$0.047) | 0.164 | 0.012 ($\pm$0.038) | 0.110 | 0.006 ($\pm$0.034) | 0.123 | 0.002 ($\pm$0.028) |
| $u$ | - | 0.001 ($\pm$0.023) | - | 0.003 ($\pm$0.042) | - | 0.003 ($\pm$0.047) | - | 0.004 ($\pm$0.040) | - | 0.010 ($\pm$0.036) |

Table 4.5. Pearson's correlation coefficients between each pair of latent representations across models on SFU+Yelp. As the content space is 62-dimensional, we compute the correlation coefficient between each dimension and report the mean and the standard deviation.

We demonstrate the invariance of our models' polarity and uncertainty representations in Table 4.5. While the ELBO objective alone results in highly covariant polarity and uncertainty latent distributions on SFU+Amazon (0.706), this drops significantly under INF (0.200) with additional reduction contributed by the ADV and MIN objectives (0.159). Interestingly, the lowest correlation between polarity and uncertainty latents on SFU+Yelp is achieved by the ELBO objective alone, although the correlations are low for all other models as well. This may be due to the ELBO-only model using the content latent only and keeping the polarity and uncertainty latents relatively random.

### 4.6.2 Evaluation of Reconstructions

#### Faithfulness and Fluency

Table 4.6 reports the self-BLEU and perplexity for each disentanglement objective. Example reconstructions are given in Table 4.7. These results show that the models are quite consistent regarding content reconstruction on the train set, but this consistency drops on dev and test. While the ADV and MIN objectives provide disentanglement gains over INF, the BLEU scores betray a possible trade off of slightly poorer content preservation, despite better perplexities.

While self-BLEU indicates the consistency of the reconstructions with respect to content, it does not necessarily indicate consistency of the reconstructions with respect to polarity and uncertainty, which often differ from their opposite value counterparts by a single token. The consistency of the INF and INF+ADV+MIN models with respect to these factors is reported in Table 4.8. The INF objective alone is quite consistent, with re-encoded F1s of 0.830 and 0.789 for polarity and uncertainty respectively. The auxiliary objectives result in improvements, to 0.914 and 0.893, suggesting that the disentanglement gains seen in Table 4.4 and Figure 4.7 have a positive effect on the consistency of the reconstructions.

**Controlled Generation**

Table 4.9 shows the accuracy of the $\mathcal{L}_{\text{INF}}$+$\mathcal{L}_{\text{ADV}}$+$\mathcal{L}_{\text{MIN}}$ model for the controlled generation task on SFU+Amazon, split by transfer direction. We found that for both polarity and uncertainty modifying the status of the input works well in only one direction: from negated to positive, uncertain to certain. We offer the following explanation for this phenomenon.

Changing a sentence from negated to positive or from uncertain to certain generally requires the *removal* of cue tokens, while the opposite directions require their *addition*. Via linear regressions between the content representations and number of tokens, we found that the content space is highly informative of sentence length, which effectively bars the decoder from adding the required negation or speculation cue tokens[10]. These results are given in Table 4.10. A manual review of correctly and incorrectly modified sentences suggested that the decoder attempts to represent the polarity/uncertainty status by modifying tokens in the input, rather than adding or removing them, in order to satisfy the length constraint. When removal is required, the cue token is often simply replaced by new tokens consistent with the representation. The inclusion of negation/speculation cue tokens, however, only seems to occur when it is possible to change an existing token to a cue token. Examples of successful and failed transfers are given in Table 4.11.

While the content space is uninformative of polarity and uncertainty, we found that it *is* highly informative of sentence length, which hinders the decoder from adding or removing tokens to satisfy the polarity/uncertainty representations. Table 4.10 reports the $R^2$ of linear regressions of the number of tokens in the input on samples from the polarity, uncertainty, and content distributions, respectively, and examples of successful and failed transfers illustrating this phenomenon are given in Table 4.11.

Note that the transferred results have the same number of tokens as the inputs, and the decoder even repeats tokens where necessary to meet the length requirement (e.g., *fit fit* in the last successful negation example in Table 4.11). In general, the decoder seems to satisfy the value of the transferred factor by changing tokens in the input. This is clear in the last successful uncertainty example in Table 4.11, where the *certain* input is correctly changed to

[10]The tendency of VAEs to focus their representation on sentence length was also observed by [212].

| | | SFU+Amazon | | | | | SFU+Yelp | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\mathcal{L}_{\text{ELBO}}$ | +$\mathcal{L}_{\text{INF}}$ | +$\mathcal{L}_{\text{INF}}$ +$\mathcal{L}_{\text{ADV}}$ | +$\mathcal{L}_{\text{INF}}$ +$\mathcal{L}_{\text{MIN}}$ | +$\mathcal{L}_{\text{INF}}$ +$\mathcal{L}_{\text{ADV}}$ +$\mathcal{L}_{\text{MIN}}$ | $\mathcal{L}_{\text{ELBO}}$ | +$\mathcal{L}_{\text{INF}}$ | +$\mathcal{L}_{\text{INF}}$ +$\mathcal{L}_{\text{ADV}}$ | +$\mathcal{L}_{\text{INF}}$ +$\mathcal{L}_{\text{MIN}}$ | +$\mathcal{L}_{\text{INF}}$ +$\mathcal{L}_{\text{ADV}}$ +$\mathcal{L}_{\text{MIN}}$ |
| BLEU | Train | 0.590 | 0.576 | 0.574 | 0.528 | 0.522 | 0.805 | 0.786 | 0.793 | 0.786 | 0.796 |
| | Dev | 0.150 | 0.189 | 0.186 | 0.146 | 0.141 | 0.492 | 0.382 | 0.394 | 0.383 | 0.398 |
| | Test | 0.153 | 0.072 | 0.073 | 0.148 | 0.144 | 0.400 | 0.298 | 0.309 | 0.300 | 0.315 |
| PPL | Train | 123.3 | 174.3 | 173.1 | 124.9 | 127.2 | 52.7 | 54.8 | 53.3 | 54.1 | 53.7 |
| | Dev | 136.4 | 186.1 | 189.1 | 140.1 | 141.3 | 75.9 | 76.9 | 76.2 | 77.4 | 76.5 |
| | Test | 136.8 | 185.9 | 187.3 | 139.3 | 142.1 | 106.2 | 105.9 | 106.0 | 106.7 | 105.5 |

Table 4.6. Reconstruction self-BLEU and reconstruction perplexity (PPL) for each model on each data split. Perplexity is computed using GPT-2 [108].

| | SFU+Amazon | SFU+Yelp |
|---|---|---|
| Input | *going home early.* | *my jack and coke was seriously lacking.* |
| Recon | *going home later.*<br>*going home movies.*<br>*going home first.* | *top brie and lobster was seriously lacking.*<br>*my pastrami and swiss was totally lacking.*<br>*the camarones and coke was totally lacking.* |
| Input | *sometime it's just not enough volume.* | *she could not and would not explain herself.* |
| Recon | *guess it's just not enough power.*<br>*believe it works just not pleasant enough.*<br>*obviously it s just not enough control.* | *she could not and would not explain herself.*<br>*she could not and would not respond herself.*<br>*she could apologize and would not introduce herself.* |
| Input | *this was perfect for the wedding cake i created.* | *the service was great and would gladly go back.* |
| Recon | *this was perfect for the original simplicity i wanted.*<br>*this is perfect for the oven cereal i smoked.*<br>*this was great for the family headset i uploaded.* | *the service was great and would easily go back.*<br>*the service was exceptional and would probably go back.*<br>*the service was great and would certainly go back.* |

Table 4.7. Example reconstructions decoded by the INF+ADV+MI model on the test sets.

| | | SFU+Amazon | | | | | | SFU+Yelp | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $+\mathcal{L}_{\text{INF}}$ | | | $+\mathcal{L}_{\text{INF}}+\mathcal{L}_{\text{ADV}}$ $+\mathcal{L}_{\text{MIN}}$ | | | $+\mathcal{L}_{\text{INF}}$ | | | $+\mathcal{L}_{\text{INF}}+\mathcal{L}_{\text{ADV}}$ $+\mathcal{L}_{\text{MIN}}$ | | |
| Factor | Pass | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| $n$ | 1 | 0.969 | 0.965 | 0.967 | 0.959 | 0.975 | 0.967 | 0.962 | 0.939 | 0.950 | 0.960 | 0.942 | 0.951 |
| | 2 | 0.816 | 0.848 | 0.830 | 0.920 | 0.908 | 0.914 | 0.932 | 0.887 | 0.908 | 0.944 | 0.907 | 0.924 |
| $u$ | 1 | 0.959 | 0.961 | 0.960 | 0.970 | 0.981 | 0.975 | 0.975 | 0.955 | 0.965 | 0.982 | 0.950 | 0.965 |
| | 2 | 0.767 | 0.820 | 0.789 | 0.930 | 0.864 | 0.893 | 0.898 | 0.900 | 0.899 | 0.933 | 0.894 | 0.912 |

Table 4.8. Consistency of the decoder with the ground-truth values of negation and uncertainty evaluated on the test set. Pass 1 refers to the predictions from the original inputs. Pass 2 refers to the predictions from the re-encoded reconstructions. Pass 1 can be considered an upper bound on the performance of pass 2.

| Transfer direction | SFU+Amazon |
|---|---|
| pos $\rightarrow$ neg | 0.38 |
| neg $\rightarrow$ pos | 0.87 |
| cer $\rightarrow$ unc | 0.36 |
| unc $\rightarrow$ cer | 0.86 |

Table 4.9. Attribute inversion accuracies by factor and direction of transfer using the $\mathcal{L}_{\text{INF}}$+$\mathcal{L}_{\text{ADV}}$+$\mathcal{L}_{\text{MIN}}$ model on the test sets.

| Latent | train | dev | test |
|---|---|---|---|
| Negation | 0.048 | 0.057 | 0.063 |
| Uncertainty | 0.051 | 0.034 | 0.041 |
| Content | 0.901 | 0.903 | 0.900 |

Table 4.10. $R^2$ of the sentence length regression models on $z$'s sampled from INF+ADV+MIN model trained on SFU+Amazon. $R^2 = 1$ represents perfect predictive ability.

*uncertain*, but the uncertainty cue replaces the negation cue.

| | Successful | Unsuccessful |
|---|---|---|
| Neg | i received my lodge grill griddle and it is extremely well made.<br>*i got my kitchen steel corkscrew but it is* **not** *well made.* | it is light in weight and easy to clean.<br>*it is tight in light and easy to clean.* |
| | if you do **n't** have one get it now.<br>*if you must have need one used it now.* | the x s were fine until i washed them.<br>*the iphone s worked fine and i return them.* |
| | but it does **not** fit the tmo galaxy s.<br>*but it does fit fit the net galaxy s.* | used for months and it is still going strong.<br>*used for now and it is still going strong.* |
| Unc | it is light in weight and easy to clean.<br>*it is slid in sponges and* **seems** *to clean.* | i received my lodge grill griddle and it is extremely well made.<br>*i received this roasting s cooking and it is still well made .* |
| | it snaps onto the phone in two pieces.<br>*it* ***should*** *affect the phone in two ways.* | glad i found these at a reasonable price.<br>*do i have these at a good price.* |
| | but we did **n't** use it on the other one.<br>*but we* **would** *have use it on the other one.* | it broke immediately when i put it on.<br>*it was immediately when i put it on.* |

Table 4.11. Successful and Unsuccessful inversions of polarity and uncertainty. Inputs are in standard font and reconstructions are in *italics*. **bold** tokens indicate negation/speculation cues.

### 4.6.3 Discussion

Overall, we found that objectives based on adversarial learning and MI minimization resulted in disentanglement and consistency gains, but that a decent balance between variable disentanglement and reconstruction ability was obtained by a simple supervision of the latent representations. This approach has the practical benefits of being simple to implement and faster to train than models using the auxiliary objectives, which require estimating additional neural layers. A brief discussion of our results in reference to the research questions posed at the beginning of the chapter is given below.

**RQ1:** *Is it possible to learn disentangled representations of negation, uncertainty, and content?*

The results suggest that it is indeed possible to estimate a statistical model in which polarity, uncertainty, and content are disentangled latent variables according to our three desiderata outlined in Section 4.3. Specifically, Table 4.4 shows high informativeness of the polarity and uncertainty spaces across objectives, and the poor predictive ability of each latent space for non-target factors suggests independence. Figure 4.7 further suggests independence across models, with median MIG scores in the 0.4-0.6 range. Finally, the low covariances in Table 4.5 demonstrates the invariance of the latent representations to each other.

**RQ2:** *How do existing disentanglement objectives compare for this task?*

Notably, the INF objective alone results in good disentanglement according to our three desiderata, suggesting that supervision alone is sufficient for disentanglement. Still, the addition of the ADV and MIN objectives resulted in slightly more informative (Table 4.4) and independent (Table 4.5) representations. While the self-BLEU scores reported in Table 4.6 suggest that content preservation is generally maintained across auxiliary objectives, small dips are seen in those using the MIN objective. This trend also holds for perplexity, suggesting that while the MIN objective can contribute to disentanglement gains, it may result in poorer reconstructions.

This work has some limitations. First, our model does not handle polarity and uncertainty scope, but rather assumes that operators scope over the entire statement, and our model was estimated on relatively short, single-statement sentences to satisfy this assumption. The follow-up work in Section 4.7 will both expand model capacity to handle longer and more complex inputs as well as explicitly classify the polarity and uncertainty of events. Second, while our models achieved high disentanglement, they fell short on the controlled generation task. We found that this was likely due to the models memorizing sentence length, constraining the reconstructions in way that is incompatible with the addition of negation and speculation cue tokens. [212] also noticed this tendency for sentence length memorization in VAEs and future will will explore their suggested remedies, such as encoder pretraining.

## 4.7 Disentangled Representation Learning on SemMed Context

The previous sections laid a foundation for DRL of polarity, uncertainty, and propositional content. They laid out a basic VAE architecture for DRL and defined three key aspects of disentanglement –informativeness, independence, and invariance– as well as corresponding evaluation metrics. We here describe follow-up research applying the findings from these experiments to the biomedical domain on SemMed Context.

One limitation of the experiments on SFU+Amazon and SFU+Yelp was that we did not explicitly model the propositional content, leaving it to a generic, high-dimensional content variable in our generative model which essentially conflated content with the style in which it was expressed. This was done out of necessity, since the data available did not have content annotations that could be used for supervision. SemMed Context, which contains annotations of event triples as well as their polarity and uncertainty, does not suffer from this limitation. We therefore modify the generative model given in Section 4.2 to split the content latent variable into separate proposition and style variables. Our new model is depicted graphically in Figure 4.8.

A main takeaway of the experiments on SFU+Amazon and SFU+Yelp was that we need not explicitly enforce disentanglement of polarity, uncertainty, and propositional content in our model via auxiliary objectives. When estimating a VAE on SemMed Context we therefore use the ELBO+INF objectives with supervision signals coming from the polarity, uncertainty,

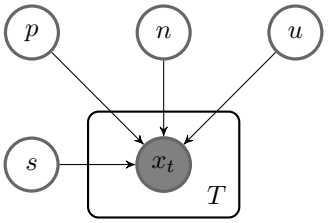



Figure 4.8. Graph of the generative model. $n$ is polarity (negation) status, $u$ is uncertainty, $p$ is propositional content, and $s$ is style.

and predicate type tasks. However, preliminary experiments applying the LSTM-based VAE model to SemRep Context failed to generate suitably fluent reconstructions. We hypothesized that this was due to the greater complexity of the biomedical language than that of the review data used in our original experiments. Biomedicine realises a sublanguage that mirrors the complexity of its subject matter: it is syntactically complex and contains nested relations as well as a highly variable vocabulary [50]. To address this increase in language complexity we refactor our original VAE to utilize a more powerful transformer-based encoder and decoder model, BART, in place of the LSTMs [110]. The BART model is larger than the LSTM meaning greater model capacity, and we utilize a model pretrained on PubMed data[11] for greater domain specificity.

The following sections propose two variations of the VAE model and describe experiments applying these to SemMed Context. The first model, SubjVAE (Section 4.7.2), uses Subjective Logic Beta and Dirichlet distributions in place of the Gaussian distributions common in VAE architectures and shows how these distributions are a better match than Gaussians for modeling discrete variables. The second model, FactVAE (Section 4.7.3), builds upon SubjVAE to learn a single Factuality latent SL Beta that encodes both polarity and uncertainty. By jointly modeling polarity and uncertainty via a single SL Beta distribution this model could be used to directly construct knowledge graphs from text such as that constructed in Chapter 3, Section 3.4.

We begin with a generic description of our BART-based VAE model which underlies both SubjVAE and FactVAE. We then describe SubjVAE and FactVAE in turn, alongside experimental results applying these models to SemMed Context.

### 4.7.1 BART as a Variational Autoencoder

BART (Bi-directional Auto-Regressive Transformer) is an encoder-decoder model based on the transformer architecture introduced in [91]. The encoder outputs token representations while the decoder predicts the next token given the preceding using cross attention between the preceding tokens and the input representations.

A key difficulty in adapting BART into a VAE is encouraging it to utilize the latent representations. In contrast to an RNN-based VAE, which uses the last hidden state of the encoder as input to the decoder, BART computes cross-attention between the output and each token

[11] `https://huggingface.co/Kevincp560/bart-large-cnn-finetuned-pubmed`

of the input. One possible solution is to simply concatenate the latent representations to the encoder token representations, in theory allowing the model to attend to both input tokens and latents. However, this strategy is prone to *posterior collapse*, a phenomenon in which the model learns an uninformative latent representations so that the model uses only the token representations when decoding, rendering the latent distributions useless [213]. Another approach is to not input the token representations to the decoder at all, instead computing cross attention between the output and the latent representations. One previous work utilized this approach, using a RoBERTa encoder to compute the latent representations which were input to a BART decoder [214]. This differs somewhat from RNN-based VAEs which typically initialize the decoder with a concatenation of the latent values rather than computing attention over them. Preliminary experiments with this latter approach showed that by including a sufficiently large style latent variable resulted in acceptable generation fluency without posterior collapse.

A definition of our basic BART-based VAE is given in Equation (4.17).

$$\boldsymbol{h} = \text{BART}_{enc}(\boldsymbol{x}) \tag{4.17}$$
$$\boldsymbol{\phi} = \text{Linear}([h_{e1}; h_{e2}]) \tag{4.18}$$
$$\boldsymbol{z} \sim p_\phi(Z|X) \tag{4.19}$$
$$\hat{y} = \text{Linear}(\boldsymbol{z}) \tag{4.20}$$
$$\hat{\boldsymbol{x}} = \text{BART}_{dec}(\boldsymbol{z}) \tag{4.21}$$

The hidden token representations $\boldsymbol{h}$ are obtained from the BART encoder applied to the input token ids $\boldsymbol{x}$. The hidden representations of the two entities in question (e.g., subject and object) are concatenated and passed through a linear layer to obtain the latent distribution parameters $\boldsymbol{\phi}$. Latent values $\boldsymbol{z}$ are sampled from the latent distribution and used to both predict the target label $\hat{y}$ and as input to the decoder to obtain output token predictions $\hat{\boldsymbol{x}}$. The two models we propose here, SubjVAE and FactVAE, vary only in the definitions of $p_\phi$ and how the latent spaces are structured, which we describe in the following sections. Like the VAE from Section 4.2, both models use the ELBO and INF objectives from Equation (4.9) and Equation (4.10), respectively.

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \lambda_{\text{INF}}\mathcal{L}_{\text{INF}} \tag{4.22}$$

where $\mathcal{L}_{\text{ELBO}}$ is computed over all latent distributions (polarity, uncertainty, predicate, and style) and $\mathcal{L}_{\text{INF}}$ is computed over the latent distributions with direct supervision (polarity, uncertainty, and predicate).

We use identical evaluation setups for SubjVAE and FactVAE on SemMed Context. Each model is trained for a total of 5 epochs with a batch size of 8 and a learning rate of $2 \cdot 10^{-5}$, and we keep the model checkpoint from the epoch with the lowest loss on the validation split.

### 4.7.2 SubjVAE

In a standard VAE which uses Gaussian distributions to model the latent variables, a feed-forward layer is used to project the latent values $z$ into the label space. In these models, the latent distribution is generally of a higher dimension than the labels, but even in the case where the latent dimensionality matches the labels it is necessary to transform $z$ into the label space via a feed-forward layer due to the difference in scale. This was the case in our original VAE model above, where 2-dimensional Gaussians were used to model the binary polarity and uncertainty tasks. Being 2-dimensional aided interpretability, but the values still needed re-scaling into $[0, 1]$ to be used for prediction. Other works on representation learning with VAEs have used smoothed discrete distributions such as HardKumaraswamy [215] or Gumbel-Softmax [216], which approximate the Bernoulli and Categorical distributions, respectively, while still being differentiable via their own unique versions of the reparameterization trick. The $z$ values sampled from these distributions can be used directly for label prediction and are thus more interpretable than Gaussians making them a good choice for modeling discrete variables.

However, there is another option for modeling discrete variables in VAEs. Instead of post-hoc transformations of discrete to continuous distributions (such as HardKumaraswamy and Gumbel-Softmax), we can model discrete variables using their already continuous conjugate prior distributions. That is, we can model binary variables with a Beta distribution (conjugate prior to the Bernoulli) and $n$-ary variables with the Dirichlet (conjugate prior to the Categorical). The domain of these distributions matches the label space (e.g., $[0, 1]$ for the Beta) so their samples can be used for prediction, and the fact that they are continuous means that they are in theory differentiable. While differentiable sampling from these distributions is not as straightforward as the reparameterization trick for Gaussians, the research in [217] defines a generic reparameterization trick for a number of continuous distributions including the Beta and Dirichlet, called implicit reparameterization gradient sampling.

One potential criticism of using Beta and Dirichlet distributions is that their parameters are not interpretable in reference to the task. The parameters – $(\alpha, \beta)$ for the Beta and $\boldsymbol{\alpha}$ for the Dirichlet– are comprised of positive integers often understood as pseudo-evidence counts for each label. An ML model that learns these parameters using gold-standard labels and cross-entropy loss as a supervision signal would increase the parameter value corresponding to the correct label at each update step until training ceased, resulting in artificially high pseudo counts[12]. However, the reparameterization of the Beta and Dirichlet distributions according to Subjective Logic, as described previously in Section 3.4.1, uses the parameters $(b, d, u) \in [0, 1]$, which are a better conceptual fit. We therefore propose SubjVAE, a VAE that uses Subjective Logic Beta and Dirichlet latent distributions and demonstrates its usefulness for modeling discrete variables on the polarity, uncertainty, and predicate type tasks in SemMed Context. The latent space definition is given in Equation (4.23), modifying Equation (4.17).

[12]To see why this is the case, consider the Beta distribution with expectation computed as $\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta}$. It follows that $\lim_{\alpha\to\infty} \mathbb{E}[X] = 1$, so, assuming a deterministic true distribution $p(X{=}1){=}1$ (e.g., a gold-standard label) and a predicted Beta distribution $q$, cross-entropy is minimized as $\alpha$ increases, i.e. $\lim_{\alpha\to\infty} -\mathbb{E}_p[\log q] = 0$. The same holds for the $\beta$ parameter under $p(X{=}0){=}1$.

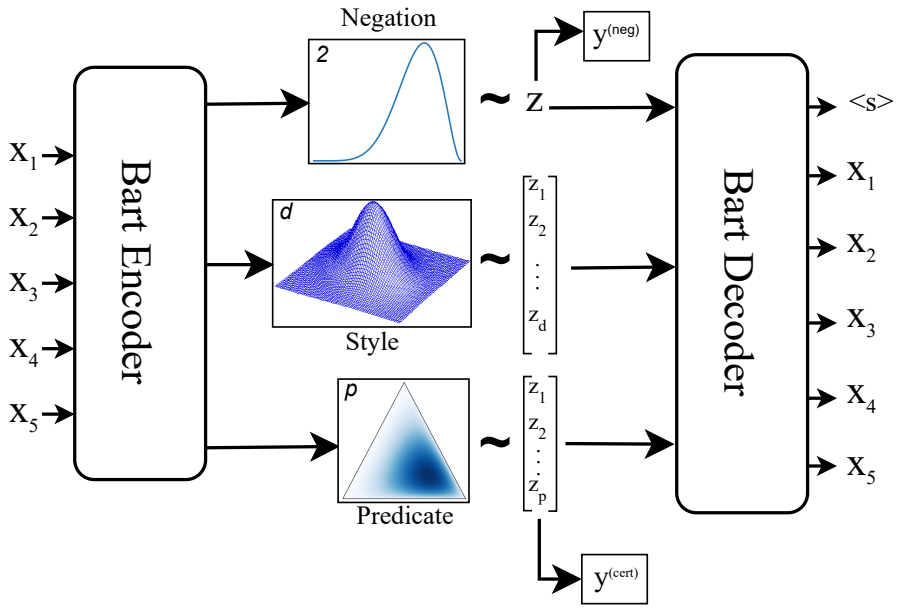



Figure 4.9. An illustration of SubjVAE. The BART encoder outputs a latent distribution for each task. For SemMed Context these are Betas for polarity and uncertainty (uncertainty not shown for space reasons), an SL Dirichlet for predicate type, and a multivariate Gaussian for the text style. As in a standard VAE, these distributions are sampled using a deterministic sampling method to obtain latent values $z$. The latent values are input to the BART decoder, which predicts the next token from the preceding (e.g., $x_1$ from `<s>`) with cross attention between the preceding tokens and the latent values.

$$z^{(n)} \sim \text{SLBeta}(b^{(n)}, d^{(n)}, u^{(n)}) \tag{4.23}$$

$$z^{(u)} \sim \text{SLBeta}(b^{(u)}, d^{(u)}, u^{(u)}) \tag{4.24}$$

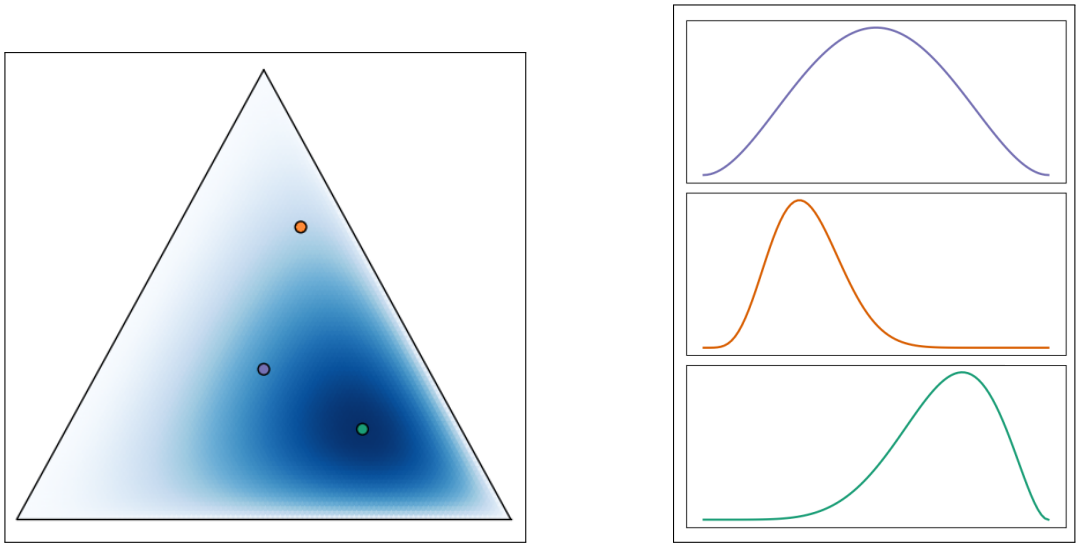
$$\boldsymbol{z}^{(p)} \sim \text{SLDirichlet}(\boldsymbol{b}^{(p)}, u^{(p)}) \tag{4.25}$$

$$\boldsymbol{z}^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{4.26}$$

Where $n, u, p$, and $s$ denote the polarity (negation), uncertainty, predicate, and style latents, respectively. The polarity and uncertainty latents are SL Beta distributions, the predicate latent is an SL Dirichlet distribution, and the style latent is a multivariate Gaussian. The parameters from each distribution are predicate as in Equation (4.17), e.g., $b, d, u = \text{softmax}(\text{Linear}([h_{e1}; h_{e2}))$.

Practically, our implementation extends the PyTorch `distributions` module[13] which implements Beta and Dirichlet distributions as well as implicit reparameterization gradient sampling. Our extensions compute the SL parameters from the standard distribution parameters (e.g., $b, d, u$ from $\alpha$ and $\beta$) via the equations given in Section 3.4. These refactored SL distributions can then be used as drop-in replacements in our BART-based VAE. An illustration of the SubjVAE architecture is given in Figure 4.9.

[13] `https://pytorch.org/docs/stable/distributions.html`



(a) Heatmap visualization of a Dirichlet distribution with points representing three samples.

(b) Beta PDFs corresponding to the points in Figure 4.10a.

Figure 4.10. Illustration of a Dirichlet prior with samples corresponding to Beta distributions.

### 4.7.3 FactVAE

SubjVAE predicts separate SL Beta distributions for polarity and uncertainty. While this approach is justified because the distributions' dimensionality matches the labels, it does not entirely accord with our definition of factuality in Definition 1. Therefore, instead of predicting separate distributions for each task, FactVAE predicts a single SL Beta distribution that can be used to predict both polarity and uncertainty. In contrast to SubjVAE, where latent distributions were sampled to obtain latent values $z$ for each task, FactVAE samples a distribution from another distribution. That is, we sample the parameters $b, d, u$ of an SL Beta from a prior distribution. Since $b + d + u = 1$, the parameters lie on the $\triangle^2$ simplex, allowing us to sample them from a 3-dimensional SL Dirichlet distribution. This notion is illustrated using the barycentric triangle visualization of binomial opinions as in Figure 4.10, where each point corresponds to an SL Beta PDF.

The core of FactVAE is therefore a latent Dirichlet distribution that represents the space of possible event factuality. Sampling from this distribution produces the parameters of an SL Beta distribution representing the factuality of a given example, which can be used to predict the polarity and uncertainty labels of that event. Specifically, polarity is predicted from the mode of the Beta distribution and uncertainty is predicted from the $u$ parameter[14]. An illustration of the FactVAE architecture is given in Figure 4.11 and the latent structure is given in Equation (4.27).

[14]Because $u \in [0, 1]$ it can be used directly to predict Certainty. However, we predict an uncertain label using a $u \geq 0.2$ threshold, as the standard $u \geq 0.5$ threshold over-emphasizes the variance in the predicted distribution.

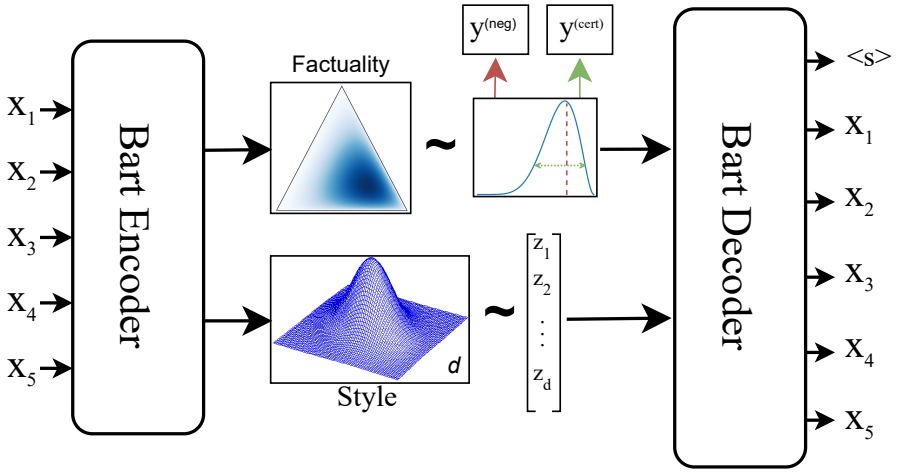



Figure 4.11. An illustration of FactVAE. The BART encoder outputs a 3-dimensional Dirichlet, which is sampled to obtain the $b, d, u$ parameters of an SL Beta PDF representing the event factuality, which is used to predict the polarity and uncertainty labels. The encoder also outputs a Gaussian distributions for style and SLBetas or SLDirichlets for any other discrete generative factors (e.g., predicate type is modeled using an SLDirichlet as in SubjVAE, but is not shown for space reasons), which are sampled to obtain latent vectors $\boldsymbol{z}$. The latent values $b, d, u$, and $\boldsymbol{z}$ are input to the BART decoder, which predicts the next token from the preceding (e.g., $x_1$ from `<s>`) with cross attention between the preceding tokens and the latent values.

$$b'^{(f)}, d'^{(f)}, u'^{(f)} \sim \text{SLDirichlet}(\boldsymbol{b}^{(f)}, u^{(f)}) \tag{4.27}$$

$$P^{(f)} := \text{SLBeta}(b'^{(f)}, d'^{(f)}, u'^{(f)}) \tag{4.28}$$

$$z^{(n)} = \text{argmax}_{\boldsymbol{z}'}\, P^{(f)}(\boldsymbol{z}') \tag{4.29}$$

$$z^{(c)} = u'^{(f)} \tag{4.30}$$

$$\boldsymbol{z}^{(p)} \sim \text{SLDirichlet}(\boldsymbol{b}^{(p)}, u^{(p)}) \tag{4.31}$$

$$\boldsymbol{z}^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \tag{4.32}$$

where, as before, $\boldsymbol{b}^{(f)}, u^{(f)} = \text{softmax}(\text{Linear}([h_{e1}; h_{e2}))$.

### 4.7.4 Evaluation

We evaluate disentanglement as before: informativeness is measured via the MI between each latent space and generative factor as well as by the predictive ability of each latent space; independence is measured via the MIG; invariance is measured via the Pearson's correlations coefficient between each pair of latent variables. We also evaluate the models' text generation ability via self-BLEU and perplexity of the reconstructions. Finally, we are able to evaluate the consistency of the reconstructions with respect to the content of the events by checking that the target subject and entities in question appear in the output.

| Latent | Polarity | | | | Uncertainty | | | | Predicate | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MI | P | R | F1 | MI | P | R | F1 | MI | P | R | F1 |
| Polarity | **0.037** | **0.693** | **0.919** | **0.763** | 0.004 | 0.533 | 0.532 | 0.532 | 0.009 | 0.082 | 0.085 | 0.046 |
| Uncertainty | 0.001 | 0.501 | 0.515 | 0.113 | **0.254** | **0.929** | **0.938** | **0.934** | 0.026 | 0.091 | 0.092 | 0.029 |
| Predicate | 0.000 | 0.505 | 0.626 | 0.385 | 0.004 | 0.544 | 0.618 | 0.470 | **0.219** | **0.922** | **0.923** | **0.922** |
| Style | 0.000 | 0.528 | 0.878 | 0.513 | 0.001 | 0.633 | 0.783 | 0.649 | 0.008 | 0.345 | 0.441 | 0.354 |

Table 4.12. Informativeness results of SubjVAE on the validation set, including the mutual information (MI) between the latent and the ground-truth labels and the macro-averaged (P)recison, (R)ecall, and F1 score.

#### Informativeness

The informativeness results of SubjVAE and FactVAE on SemMed Context are given in Table 4.12 and Table 4.13, respectively. In both models, we see that each latent space is highly informative of its target factor and non-informative of all others. Additionally, the style latent is a poor predictor of all supervised generative factors with low MIs and F1 scores suggesting that it is used only to determine the general style of the output and not any of the predication content.

| Latent | Polarity | | | | Uncertainty | | | | Predicate | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MI | P | R | F | MI | P | R | F | MI | P | R | F |
| Factuality | **0.022** | **0.715** | **0.905** | **0.780** | **0.180** | **0.940** | **0.931** | **0.935** | 0.025 | 0.078 | 0.094 | 0.033 |
| Predicate | 0.000 | 0.505 | 0.617 | 0.394 | 0.002 | 0.544 | 0.617 | 0.469 | **0.222** | **0.934** | **0.929** | **0.931** |
| Style | 0.000 | 0.530 | 0.882 | 0.519 | 0.002 | 0.646 | 0.804 | 0.669 | 0.009 | 0.367 | 0.473 | 0.382 |

Table 4.13. Informativeness results of FactVAE on the validation set, including the mutual information (MI) between the latent and the ground-truth labels and the macro-averaged (P)recison, (R)ecall, and F1 score.

#### Independence

The MIG values for SubjVAE and FactVAE for each latent space are shown in Table 4.14. We first note that, because the MIG is averaged over the entropy of of the generative factor (cf. Equation (4.16)), larger label spaces are penalized more. Thus the MIG for the 13-dimensional predicate is much lower than the MIGs for the binary polarity and uncertainty labels. Both models achieve respectable MIG values, although FactVAE achieves lower MIGs for polarity and uncertainty than SubjVAE, despite higher predictive performance as shown in Table 4.13. Nevertheless, the MIGs for FactVAE suggest independence of each generative factor despite polarity and uncertainty being modeled by a single latent distribution.

| | SubjVAE | | FactVAE | |
|---|---|---|---|---|
| Polarity | 0.619 | ($\pm$0.011) | 0.383 | ($\pm$0.006) |
| Uncertainty | 0.747 | ($\pm$0.004) | 0.535 | ($\pm$0.001) |
| Predicate | 0.079 | ($\pm$0.002) | 0.086 | ($\pm$0.002) |

Table 4.14. The Mutual Information Gap (MIG) for SubjVAE and FactVAE for each generative factor. Note that the MIG for polarity and uncertainty are both computed from the factuality latent. The reported results are averaged over 10 runs.

#### Invariance

We report the Pearson's correlation coefficients between each latent space in Table 4.15 and Table 4.13. Generally, the low correlations show that the latent spaces are invartiant to each other, and FactVAE has lower correlations than SubjVAE.

| | Polarity | Predicate | Style |
|---|---|---|---|
| Uncertainty | -0.002 | -0.006 | -0.006 |
| Polarity | - | -0.007 | -0.001 |
| Predicate | - | - | 0.000 |

Table 4.15. Invariance results on the validation set, measured as the Pearson's correlation between samples from each latent space.

| | Predicate | Style |
|---|---|---|
| Factuality | 0.001 | 0.000 |
| Predicate | - | 0.000 |

Table 4.16. Invariance results on the dev set.

#### Generation

The self-BLEU and Perplexity (PPL) results are given in Table 4.17. Perplexity is computed under GPT-Neo [218] fine-tuned on scientific text for better domain-specificity[15]. We see that, compared to the LSTM-based VAE from the previous experiments, SubjVAE and FactVAE are better able to generalize: while the self-BLEU on the train split is lower than the LSTM VAEs on both SFU+Amazon and SFU+Yelp, it is better on than those models on the validation and test splits (cf. Table 4.6). This can likely be attributed to the larger model capacity of BART as well as the domain specific pre-training. Regarding perplexity, we see that the fluency of the model's reconstructions is very consistent across data splits, at about 62 for each. Comparing these values to those on the SFU data (cf. Table 4.6), we see that SubjVAE and FactVAE achieve perplexities $\frac{1}{2}$ to $\frac{1}{3}$ of the LSTM model on SFU+Amazon and comparable to the perplexities on SFU+Yelp, despite considerably greater complexity of the biomedical text in SemMed Context.

| | SubjVAE | | FactVAE | | |
|---|---|---|---|---|---|
| | self-BLEU | PPL | self-BLEU | PPL | PPL (inputs) |
| Train | 0.241 | 62.4 | 0.259 | 62.0 | 40.6 |
| Val | 0.196 | 62.2 | 0.201 | 62.3 | 37.6 |
| Test | 0.196 | 62.4 | 0.202 | 62.2 | 37.7 |

Table 4.17. Reconstruction (BLEU) and perplexity (PPL) of the text generated on each data split for SubjVAE and FactVAE. Perplexity is computed using GPT-Neo [218] fine-tuned on scientific text. For comparison with the reconstructions, PPL (inputs) is the perplexity of the gold-standard input text computed under GPT-Neo.

We provide example reconstructions in Table 4.18 (SubjVAE) and Table 4.19 (FactVAE). As the perplexity values suggested, we see that the fluency of the reconstructions is quite high and they generally match the input in both style and content. However, we see that the

[15] `https://huggingface.co/Abirate/gpt_3_finetuned_multi_x_science`

specifics are inconsistent, with the reconstruction often using different entities altogether. For example, the last reconstruction in Table 4.18 does not correctly reconstruct any of the gene names, "*CYP2C8, CYP3A4 and CYP5A6*". In the SFU experiments we were not able to perform a more precise evaluation of the semantic consistency of the reconstructions with the input since we had no explicit signal of the content variable and so fell back on the self-BLEU score as an approximate metric of consistency. In SemMed Context, however, we have signals of the subject and object entities. Using simple string matching it is possible to compute the percentage of examples whose reconstruction contains both entities.

| | |
|---|---|
| Input | *Sublingual route has been widely used to deliver small molecules into the bloodstream and to modulate the immune response at different sites.* |
| Recon | *Sublingual delivery has been widely used to deliver small molecules to the site and to differentially modulate the immune system at different times in vivo.* |
| Input | *In conclusion, systemic pilocarpine produced oxidative brain damage, whereas local pilocarpine brain injection had no effects.* |
| Recon | *In conclusion, local pilocarpine produced systemic oxidative brain tissue damage, whereas systemic pilocaprine injected injected i.c.v. did not produce these effects.* |
| Input | *These results directly link defective 2'-O-methylation of the tRNA anticodon loop to FTSJ1 mutations, suggest that the modification defects cause NSXLID, and may implicate Gm34 of tRNA(Phe) as the critical modification.* |
| Recon | *These results suggest that incorrect modification of the TGF-beta1-induced XRCC2 mutants directly interfere with the N terminus of tRNA(S235D), and point to the structural defects of tRNAs, implicating tRNA modification as one possible mechanism for defective tRNA methylation.* |
| Input | *H2S could not influence the activities of CYP2B6, CYP2D6 and CYP2C19 in rats, while H2S could induce the activity of CYP3A4 and inhibit the activity of CYP2C9 in rats.* |
| Recon | *H2S could not influence the activities of CYP2C8, CYP3A4 and CYP5A6 in the activity of HSCs, and B6 could inhibit the activity in H2S induced hypertension while B2S did not influence HSC activity of CYC2C19 cells.* |

Table 4.18. Example reconstructions decoded by SubjVAE on the validation set.

For SubjVAE on the validation set, 74% of reconstructions contained at least one target entity, 30% contained both, and 28% contained the correct *subject-predicate-object* 3-tuple. For FactVAE, 76% contained at least one target entity, 32% contained both, and 31% expressed the correct *subject-predicate-object* 3-tuple. Overall, these results indicate that the reconstructed text of the these models lacks consistency with the input with respect to the event content. While decoder consistency was not the primary goal of these experiments (we were focused on disentanglement of the latent representations) it points to potential improvements to be made in future work: inclusion of the entities as variables in the generative model and training the decoder with an additional consistency objective, such as the wake-sleep procedure in [190].

#### Prediction Results on Test Split

We report the prediction performance of SubjVAE and FactVAE on the SemMed Context test split in Table 4.20. Both models improve over the baseline BERT model from Section 3.3 on

| | |
|---|---|
| Input | *Sublingual route has been widely used to deliver small molecules into the bloodstream and to modulate the immune response at different sites.* |
| Recon | *Subcutaneous delivery has been widely used to deliver small doses to the whole body and has been shown to modulate the immune response to systemic drugs at the dose level.* |
| Input | *In conclusion, systemic pilocarpine produced oxidative brain damage, whereas local pilocarpine brain injection had no effects.* |
| Recon | *In vivo, pilocarpine produced systemic brain oxidative brain damage, whereas local pilocarib injection resulted in oxidative brain injury.* |
| Input | *These results directly link defective 2'-O-methylation of the tRNA anticodon loop to FTSJ1 mutations, suggest that the modification defects cause NSXLID, and may implicate Gm34 of tRNA(Phe) as the critical modification.* |
| Recon | *These results further implicate the TGF-beta2 -induced mutational modification of Jaz2(SNP43) residues directly causes the defective TGF -beta1->TGF -alpha complex, and suggest that incorrect mutational assignment of the 5' UTRs may cause the defects of tRNA, leading to autoimmunity.* |
| Input | *H2S could not influence the activities of CYP2B6, CYP2D6 and CYP2C19 in rats, while H2S could induce the activity of CYP3A4 and inhibit the activity of CYP2C9 in rats.* |
| Recon | *H2S could not influence the activities of CYP2C19, CYP3A4, CYH6A2 and CYP4A in rats, while H2S and BH2 could inhibit the activities and could induce the activity of CYS2C6 in the presence of DZN.* |

Table 4.19. Example reconstructions decoded by FactVAE on the validation set.

all three tasks, with gains of 12 - 15 percentage points. Furthermore, we see that FactVAE slightly outperforms SubjVAE on the test split across tasks.

| | Polarity | | | | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|
| | Negative | | | Positive | | | Macro Avg. | | |
| Model | P | R | F | P | R | F | P | R | F |
| Baseline | 0.462 | 0.517 | 0.488 | 0.995 | 0.994 | 0.995 | 0.795 | 0.778 | 0.785 |
| SubjVAE | 0.503 | 0.781 | 0.612 | 0.998 | 0.993 | 0.995 | 0.924 | 0.927 | 0.925 |
| FactVAE | 0.540 | 0.723 | 0.618 | 0.997 | 0.994 | 0.996 | 0.934 | 0.931 | 0.932 |

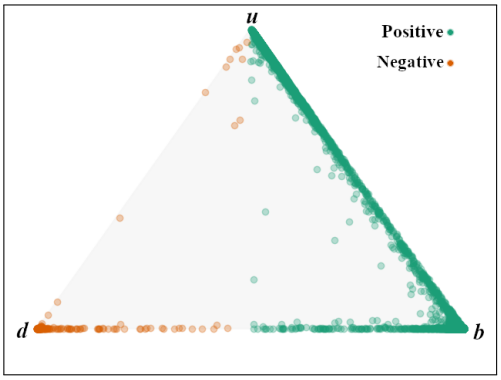
| | Uncertainty | | | | | |
|---|---|---|---|---|---|---|
| | Uncertain | | | Certain | | |
| Model | P | R | F | P | R | F |
| Baseline | 0.720 | 0.821 | 0.767 | 0.979 | 0.962 | 0.970 |
| SubjVAE | 0.885 | 0.888 | 0.886 | 0.987 | 0.986 | 0.987 |
| FactVAE | 0.893 | 0.885 | 0.889 | 0.986 | 0.988 | 0.987 |

Table 4.20. Results of SubjVAE and FactVAE on the SemMed Context test split, compared to the baseline BERT model from Section 3.3.

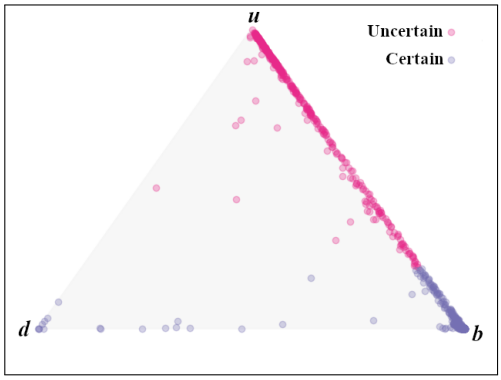
### 4.7.5 Discussion

#### Additional Results for FactVAE

One of the main goals of FactVAE is to model factuality with a distribution that theoretically accords with our definition of factuality given in Definition 1, towards RA1. We here provide visualizations of the learned distributions as evidence of its success.




(a) Scatter plot of samples from the latent SL Dirichlet distribution, colored by Negation value.




(b) Scatter plot of samples from the latent SL Dirichlet distribution, colored by Certainty value.

Figure 4.12. Illustration of a Dirichlet prior with samples corresponding to Beta distributions.

First, Figure 4.12 plots samples from the latent factuality SLDirichlet distribution corresponding to examples in the validation set, colored by polarity and uncertainty value. We see that most values fall on the edges of the simplex and that they primarily congregate at the vertices, meaning that at least one of the parameters $(b, d, u)$ is 0. This is likely due to the nature of the labels in SemMed Context: both polarity and uncertainty are assigned binary labels, meaning that a well-estimated model will make predictions near the edges/vertices in order to reduce the cross-entropy loss as much as possible. Further, samples in the middle of the space indicate second-order uncertainty of the model regarding its predictions, which is rare under cross-entropy without any explicit calibration of model uncertainty. The following subsection Section 4.7.5 discusses a possible remedy for this.

Each of the points in Figure 4.12 is a specific set of $(b, d, u)$ values that define an SL Beta distribution, which further represents the factuality of the example. Figure 4.13 shows plots of SL Beta distributions for each factuality label taken from examples in SemMed Context. We see that the model is able to predict interpretable distributions for each label, such that the mode aligns with the polarity and the variance aligns with the uncertainty of the example.

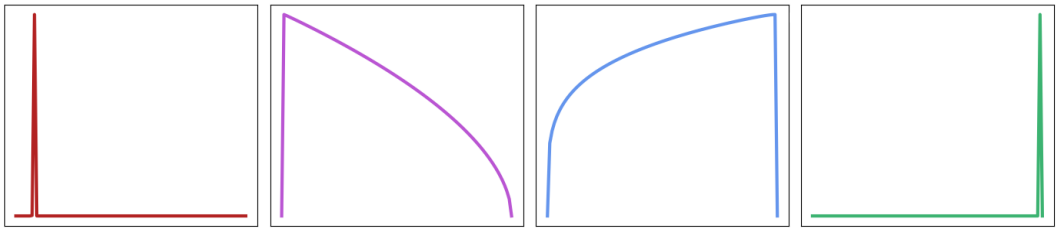


Figure 4.13. Example distributions predicted by FactVAE for CounterFact, Doubtful, Possible, and Fact events.

#### Annotation Uncertainty

The experimental results above show that both SubjVAE and FactVAE successfully disentangle polarity, uncertainty, and predicate type while also achieving respectable prediction performance compared to the baseline model described in Section 3.3. Because the latent distributions are of the same dimensionality and domain as the labels, samples from the latent space can be used directly for prediction and do not need to be transformed as is required

when using Gaussian latent distributions.

However, there is an additional benefit to using SL Beta and SL Dirichlet distributions that our experiments were not able to illustrate. These distributions, because they are conjugate priors of the Bernoulli and Categorical, encode second-order uncertainty regarding the label distribution. This means that they are capable of modeling uncertainty in the supervision signal. As is the case with most ML datasets, SemMed Context only provides a single discrete label for each example and so there is no annotation uncertainty. However, *data perspectivism* –introduced before in Section 3.4– aims to utilize continuous labels that embrace annotation uncertainty arising from individual annotators, annotation guidelines, or the task itself to build ML systems that are more representative of the real world [37]. In fact, it is possible to construct continuous labels that are themselves SL Beta or SL Dirichlet distributions, meaning that the latent distributions predicted by SubjVAE and FactVAE could be supervised directly, rather than via sampling as we did here. This aggregation of multiple discrete labels into a single continuous label encoded using Subjective Logic is called Subjective Logic Encodings (SLEs) and they are a novel method of encoding and aggregating labels that was invented by the author. However, a full discussion of SLEs would put us too far afield, so we refer the reader to Appendix A for a more in-depth treatment as well as preliminary experimental results showing their efficacy for classification tasks. In any case, future work could leverage additional annotation resources such as weak supervision or crowdsourced annotations to generate continuous labels for polarity, uncertainty, and predicate type, which could then be encoded as SLEs. This is especially relevant for the uncertainty task, as previous work found that there is, somewhat ironically, inherent uncertainty amongst individuals annotating textual uncertainty [133].

### 4.7.6 Limitations and Future Work

Due to resource constraints, we were not able to conduct as many experiments with the BART-based VAEs as we were for the LSTM-based VAEs. In the future, it would be beneficial to experiment with BART-based VAEs that use Gaussians or smoothed discrete distributions in order to make a fair comparison of SubjVAE and FactVAE with prior work. For the same reason, we were unable to completely evaluate the generation ability of SubjVAE and FactVAE, and future work should investigate their performance in controlled generation and whether they suffer from the same sentence length bias as the LSTM-based VAE (cf. Section 4.6.2). Finally, we noted the poor consistency of the decoder with respect to propositional content, despite the relatively high fluency of the reconstructions. Future research should explore the development of a latent structure that includes all members of the event, such as subject and object entities.

## 4.8 Conclusion

This chapter investigated the practicalities of learning disentangled representations of polarity, uncertainty, and event content using deep learning, following the theory that these generative factors are semantically independent. The first part of the chapter explored different disentanglement objectives using an LSTM-based VAE on general-domain data. We found that while auxiliary objectives based on mutual information and adversarial learning did result in disentanglement gains according to our metrics, simple supervision of the generative factors with gold-standard or weak labels provided respectable disentanglement and prediction performance. We then applied this result to the biomedical domain via experiments on SemMed Context. We developed a more powerful VAE model based on BART and refactored the latent space into two model variants –SubjVAE and FactVAE– which utilized Subjective Logic Beta and Dirichlet distributions in place of the Gaussians and smoothed discrete distributions (e.g., HardKumaraswamy and Gumbel-Softmax) commonly used in VAEs. By using these distributions, the models are better aligned to modeling factuality as the latent representations are both more interpretable and easier to use than Gaussians, since they can be used directly for prediction. On this point, both models achieved high predictive performance on the generative factors and respectable text generation ability, with FactVAE slightly outperforming SubjVAE on the validation and test splits, and both significantly outperformed the BERT baseline model from Section 3.3.

# Chapter 5

# Factualized Event Extraction with Interpretable Predictions

Section 1.1 and Chapter 4 argued that polarity and uncertainty are independent dimensions of event factuality. This independence was demonstrated not only from a theoretical perspective drawing on ideas from semantics, but also from a practical one using a VAE to enforce disentanglement of the polarity, uncertainty, and content representations of events in both general-domain and scientific text. A key takeaway from these experiments is that disentanglement is attainable using the task supervision signal only and additional disentanglement objectives such as those based on mutual information improved disentanglement only slightly. In the broader context of this thesis, these findings suggest that a multi-task model for FEE can simply supervise each task separately and need not explicitly enforce independence between them.

Following this notion, and in contrast to the theoretical and probabilistic bent of the previous chapter, this chapter specifically targets multi-task FEE performance using discriminative models based on PLMs. Towards this aim, we propose Levitated Context Markers (LCMs), a novel method for FEE which builds off of previous approaches for NER and RE. Informed by the findings above, our model is able to jointly predict event type, polarity, and uncertainty status using only standard cross-entropy loss. In addition to obtaining performance improvements over a strong baseline model, LCMs utilize a sparse attention mechanism which can detect context cue spans in an unsupervised fashion, lending interpretability to the model predictions.

This chapter is split into two parts. Section 5.1 *Levitated Context Markers* is an adaptation from research already published by the author in the Journal of Biomedical Informatics [219] which establishes the usefulness of LCMs on the Contextualized Medication Event Dataset (CMED) conducted as part of the 2022 n2c2 shared task [134]. Despite the favourable results of this work, our approach had three main limitations. First, the relatively small size of CMED made it difficult to estimate the models as well as evaluate their generalizability. Second, the approach estimated separate models for EE and each context dimension, rather than attempting joint FEE as we aim for in this thesis. Third, the events in CMED are comprised of a single argument, meaning our approach was not evaluated on multi-argument event schemas, which are more prevalent in the research literature. Section 5.2 *LCMs for Multi-*

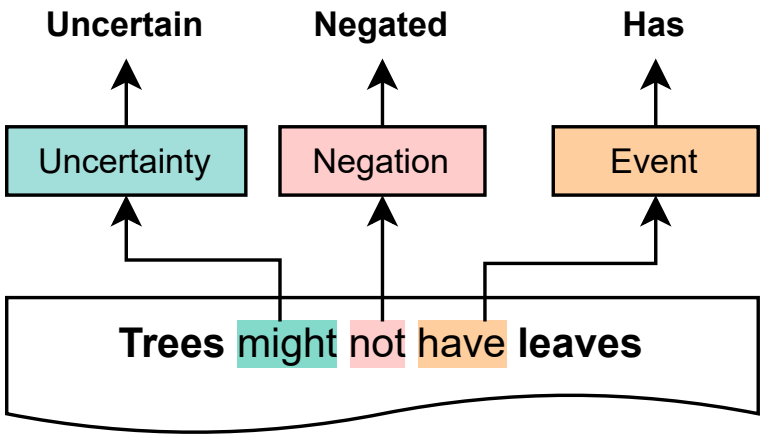



Figure 5.1. An illustration of our proposed model. An input sentence is passed to three different heads for predicting the event, negation, and uncertainty, respectively. Each head makes its prediction based only off of the relevant subspans in the input. For example, the Negation head would make its prediction using the two entity spans *"Trees"* and *"leaves"* as well as the negation cue *"not"*.

*Task FEE* attempts to address these limitations by 1) applying LCMs to the large amount of data in SemMed Context, 2) experimenting with different architectures for joint learning of events and polarity/uncertainty, and 3) extending LCMs to the more complex event schema used in SemMed Context.

## 5.1 Levitated Context Markers

Our main goal is to estimate a model that is able to correctly predict contextual information regarding a given event. A standard approach to this, which we have already employed with our baseline model in Chapter 3 and FactVAE in Chapter 4, is to add a task-specific prediction head to a generic input encoder such as BERT. Under this approach, the task head will receive all of the information in the input to make a prediction. It is important to note, however, that heads for different tasks will likely be more or less concerned with different parts of the input. We illustrate this in Figure 5.1, using the simple example from Chapter 4. In the claim *"Trees might not have leaves"* the event type head ought to make its prediction given *"have"*, the negation head given *"not"*, and the uncertainty head given *"might"*. In other words, the task-specific heads ought to be concerned not only with making correct predictions, but also with identifying the relevant cue spans from the input in order to make the predictions more interpretable.

LCMs do precisely that. They are a task-specific layer which both predicts the target label and automatically detects relevant cue spans given only the output of a generic input encoder. LCMs achieve this via a context function which aggregates the input token representations in a way that is useful for the target task, automatically selecting relevant cue spans utilizing a sparse attention mechanism. Our experiments show that LCMs not only outperform a strong baseline model for both EE and CC, but also that the corresponding attention weights highlight semantically relevant cue spans, lending interpretability.

### 5.1.1 Background

This section provides a brief summary of related research on automatic cue discovery as well as necessary background on levitated markers, the precursor to LCMs.

#### Cue Discovery

Previous works on detecting context cues are almost exclusively based on supervised approaches. For example, [220] trained a conditional random field with word and shallow parse features on the CoNLL 2010 shared task on speculation cue detection, and [116] used similar features to estimate a decision tree for speculation cue detection as well as a similar model for negation [115]. Later approaches followed suit with updated models such as RNNs (e.g., [120]) and PLMs (e.g., [125]). In contrast to the ML based approaches, others utilized curated cue word lists. For example, the system in [221] uses a list of speculation cues with manual supplementation from a small set of documents, incorporating rules based on syntactic patterns that indicate uncertain statements.

#### Explainable AI

Despite the lack of works directly addressing cue discovery, it can be related to the task of rationale extraction (also known as feature attribution) within Explainable AI (XAI) in which models are trained to produce a reason for each prediction, generally without any explicit supervision. For example, [222] and [223] use differentiable masks to highlight a subset of the input text as a rationale for text classification. Our proposed LCMs are similar to these methods, but instead of explicitly predicting a rationale using a separate model with a specialized loss function, LCMs produce the rationale as a by-product of prediction using a sparse attention mechanism with the standard cross-entropy loss. Our method thus falls within attention-based extractive rationalization techniques and is similar to [224], who propose various sparse attention mechanisms and demonstrate interpretability of the cross attention mechanism in sequence-to-sequence tasks such as textual entailment, machine translation, and summarization. In contrast to this, our approach is aimed at text classification tasks over a single sequence, and LCMs are first and foremost designed to improve classification performance while interpretability comes as a welcome side-effect.

While we find that our approach is successful at identifying relevant cues in a majority of cases (cf. Table 5.14), there is an ongoing debate as to whether attention scores are in fact suitable for explanation (see [225] for an overview). As part of this debate, [226] evaluate sparse attention of the same formulation that we use here (but without the addition of levitated markers), finding that sparsity does not necessarily increase interpretability. [227] provide a possible explanation for this, introducing the notion of *combinatorial shortcuts* which mean that attention weights can carry information more than simply importance weights. On the other hand, research has shown that attention can provide explanations in

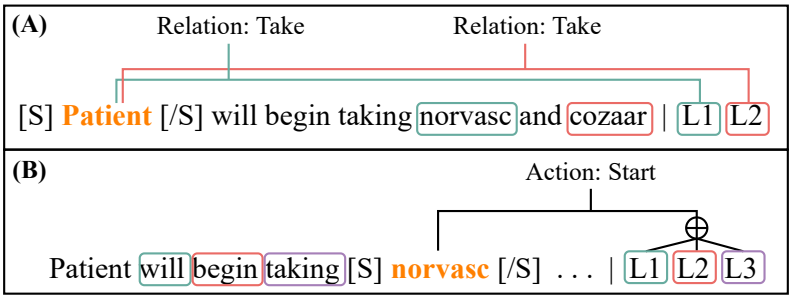



Figure 5.2. Comparison of levitated markers for their original use case of relationship extraction (A) and our task of context classification (B). In (A), a hypothetical relationship extraction task, the subject of the relation is given solid markers (`[S]`, `[/S]`) and all candidate objects are given levitated markers (e.g., `L1`, with colors indicating shared position IDs). A classifier then makes a prediction for each subject-object pair. In (B), the medication span is given solid markers and multiple spans are given levitated markers to act as the context. Instead of making a prediction for each pair, the levitated span representations are pooled and used to make a single prediction.

some cases. [228] find that self-attention in BERT encodes syntactic phenomena and [229] find that while attention is not explanation for tasks over a single input sequence, it can be explanation for paired sequence tasks. This latter finding is especially relevant to this work as we essentially transform a single sequence task into a paired sequence task (Section 5.1.3).

#### Levitated Markers

Levitated markers [102] were originally developed to speed up model inference for RE. Instead of inserting marker tokens into the input text around each subject and object span (cf. [101]), marker tokens are appended to the input and tied to corresponding input spans by a shared position ID and a directional attention mask. Follow up work [103] achieved performance gains by strategically packing together multiple levitated markers into a single training instance, in theory allowing the model to learn correlations between object spans. In essence, this encoding strategy allows a pretrained transformer to focus on multiple levitated "context" spans for a single solid-marked "target" span, without extensive augmentation of the input text. LCMs leverage this to utilize potentially many context spans for a given medication change event, pooling the context representations rather than making separate predictions for each. A comparison of our method to the previous work is given in Figure 5.2.

### 5.1.2 Data

Chapter 2 showed that EE datasets are often lacking context annotations, and datasets in the clinical domain are no exception. Towards a remedy for this, The Contextualized Medication Event Dataset (CMED) –released as part of the 2022 n2c2 shared task [137]– brings multiple context dimensions together into a single dataset, providing annotations of 5 context dimensions (Action, Actor, Certainty, Negation, and Temporality) relating to medication change events in clinical notes. CMED contains 500 clinical notes from 296 patients and was annotated for three subtasks:

- **Medication Detection** (MD): Find all medication spans in the input text.

- **Medication Change Event Extraction** (EE): For each medication mention, determine if a medication change is being discussed (Disposition) or not (NoDisposition), or if it is unclear (Undetermined).
- **Context Classification** (CC): Classify each Disposition event according to 5 context dimensions, described in Table 5.1.

| Context | Definition | Labels |
|---|---|---|
| Action | What medication change is being discussed. | Start, Stop, Increase, Decrease, OtherChange, UniqueDose, Unknown |
| Actor | Who initiated the change. | Physician, Patient, Unknown |
| Certainty | How likely the change is to occur. | Certain, Hypothetical, Conditional, Unknown |
| Negation | Whether the change is negated. | Negated, NotNegated |
| Temporality | When the change is said to occur. | Past, Present, Future, Unknown |

Table 5.1. Context dimensions along with their definitions and possible labels.

The 500 total notes are split into 350 for model training, 50 for validation, and 100 for testing. Each subtask relies on a subset of labels from the previous subtask. Therefore, while the MD task encompasses 350 training notes and 6,196 medication mentions, only 340 training notes contain any medication spans for input to the EE task, and there are only 272 notes containing 1,191 Disposition events as input to the CC task. Detailed datasets statistics are given in Table 5.2.

| Task | Label | Train | Val | Test |
|---|---|---|---|---|
| Event | NoDisposition | 4535 | 725 | 1326 |
| | Disposition | 1191 | 221 | 335 |
| | Unknown | 470 | 87 | 122 |
| Action | Start | 471 | 97 | 131 |
| | Stop | 280 | 60 | 67 |
| | UniqueDose | 263 | 22 | 88 |
| | Increase | 106 | 23 | 22 |
| | Decrease | 41 | 13 | 13 |
| | Unknown | 29 | 6 | 14 |
| | OtherChange | 1 | 0 | 0 |
| Actor | Physician | 1084 | 194 | 311 |
| | Patient | 89 | 17 | 17 |
| | Unknown | 18 | 10 | 7 |

| Task | Label | Train | Val | Test |
|---|---|---|---|---|
| Negation | NotNegated | 1163 | 217 | 329 |
| | Negated | 28 | 4 | 6 |
| Certainty | Certain | 1001 | 175 | 281 |
| | Hypothetical | 105 | 29 | 33 |
| | Conditional | 83 | 17 | 15 |
| | Unknown | 2 | 0 | 6 |
| Temporality | Past | 613 | 131 | 173 |
| | Present | 440 | 54 | 132 |
| | Future | 112 | 33 | 29 |
| | Unknown | 26 | 3 | 1 |

Table 5.2. Dataset statistics for the train, validation, and test sets for the Event Extraction and Context Classification tasks.

### 5.1.3 Model

We employ the same fundamental model for the EE and CC tasks. Input text containing the target medication mention is first preprocessed by inserting solid markers before and after the mention span. We also append levitated markers –described below– to the input, which represent task-specific context. This is then passed through a PLM encoder to obtain a hidden

representation of the medication mention and context, which is input to a classification layer for prediction. Formally,

$$\boldsymbol{h}_m = \text{PLM}(x_m) \tag{5.1}$$

$$\boldsymbol{h}_\ell = \text{PLM}(\boldsymbol{\ell}) \tag{5.2}$$

$$\hat{y} = f([\boldsymbol{h}_m;\ c(\boldsymbol{h}_\ell)]) \tag{5.3}$$

where $x_m$ is the medication mention, $\boldsymbol{\ell} = \{\ell_0, \ell_1, ..., \ell_L\}$ are the levitated markers, and $f(\cdot)$ is the classification function. The $c(\cdot)$ function pools the hidden representations of the levitated markers into a fixed-length context vector. Model estimation uses the standard cross entropy loss of the predicted class against the gold label.

The following subsection describes levitated markers and how we compute the context vector from them.

#### Levitated Context Markers

Previous work using levitated markers for relationship extraction assigned solid markers to subject spans and levitated markers to potential objects, making predictions for each subject-object pair [103]. In contrast, LCMs pool multiple levitated markers related to a single event in order to focus the model on potentially useful subspans of the input, beyond simply the medication span representation computed from the global input context. We experiment with two methods for choosing which spans to assign levitated markers.

1. Window: Mark all spans in a $W = \frac{L}{2}$ window before and after the target medication mention, where $L$ is the total number of levitated markers.
2. Rule-based: Mark spans according to some rule or based on external knowledge of the task. We choose the following rules for each task:
    - Event: we mark all verbs and auxiliary verbs in the input. Additionally, we experimented with adding specific types to verbs indicating change, using the semantic tagger PyMUSAS[1].
    - Action, Temporality: We mark all verbs and auxiliary verbs in the input.
    - Actor: We mark all nouns and pronouns in the input.
    - Certainty, Negation: We mark spans using lists of negation and speculation cues extracted from the SFU review corpus [138].

For the above rules, parts of speech are labeled using ScispaCy [230].

[1] `https://ucrel.github.io/pymusas/`

We propose three different definitions of the context function $c(\cdot)$ in Equation (5.3) to combine the levitated marker representations.

1. Max Pooling: We compute the maximum value of each dimension in the hidden representation across all levitated markers[2]. The result is a single vector of dimension $D$.

$$c(\boldsymbol{h}_\ell) := \max_{i \in L} \boldsymbol{h}_{\ell_{ij}},\ \forall j \in D \tag{5.4}$$

2. Softmax Attention Pooling: We introduce an attention mechanism between the medication span and each levitated marker. The learned attention weights are then used to compute a weighted average of the levitated marker representations.

$$c(\boldsymbol{h}_\ell) := \sum_{i=1}^{L} \alpha_{mi} \boldsymbol{h}_{\ell_i} \tag{5.5}$$

$$\boldsymbol{\alpha}_m = \boldsymbol{\rho}(e_m) = \frac{\exp(e_{mi})}{\sum_{i=1}^{L} \exp(e_{mi})},\ \forall i \in L \tag{5.6}$$

$$e_{mi} = a([\boldsymbol{h}_m; \boldsymbol{h}_{\ell_i}]) \tag{5.7}$$

where $a$ is an alignment model that outputs a score for each medication span-marker pair[3], and $\rho$ is the softmax function which projects the raw attention scores $\boldsymbol{e}_m \in \mathbb{R}^L$ into probabilities $\boldsymbol{\alpha}_m \in \triangle^{(L-1)}$.

3. Sparse Attention Pooling: We replace the softmax projection function $\rho$ in Equation (5.6) with a sparse version, Sparsegen-lin [231], which was previously used for sparse self-attention in transformer models [232].

$$\rho(e_{mi}) = \max\left\{0, \frac{e_{mi} - \tau(\boldsymbol{e}_m)}{1 - \lambda}\right\} \tag{5.8}$$

where $\tau(\cdot)$ is a thresholding function, which ensures $\sum_{i=1}^{L} \rho(e_{mi}) = 1$ and is the result of solving an optimization problem given $\boldsymbol{e}_m$ and $\lambda$[4]. The $\lambda < 1$ hyperparameter controls how many scores in $\boldsymbol{e}_m$ become 0 in the resulting probability vector. Specifically, as $\lambda \to 1$, the result approaches a one-hot vector and as $\lambda \to -\infty$ the result approaches uniform. In theory, a sparse probability vector will force the model to select only those levitated spans that are highly correlated with the downstream task. We can then inspect these probabilities at inference to gain some interpretability of the model's predictions.

### 5.1.4 Results

For the EE task, we report the average performance on the validation set across 6 different cross-validation splits. Due to the small size and imbalanced class distributions of the CC

[2]Max pooling consistently outperformed mean pooling in our preliminary experiments.
[3]We use a single linear layer $\hat{y} = \tanh([h_m; h_{\ell_i}]w^\top),\ w \in \mathbb{R}^{(2D)}$
[4]We refer the reader to Appendix A.1 of [231] for details.

data, we average results over 15 runs: 5 different cross-validation splits and 3 different random model initializations. The best performing model on the validation set is then evaluated on the test set, having been trained on the standard train split for reproducibility. We fix model hyperparameters according to our participation in the n2c2 shared task, which are detailed in 5.1.7.

#### Event Extraction

We experiment with LCMs using max pooling and the rule-based approach discussed in Section 5.1.3. Both attention pooling methods and the window-based levitated markers underperformed max pooling with rule-based levitated markers in preliminary experiments, so we omit these results for space reasons. Our baseline model uses solid markers surrounding the medication span only.

We report the micro- and macro-averaged results on the EE task in Table 5.3. While max pooled LCMs perform poorer than the baseline, adding in verb type information results in the best performance according to both the micro and macro averaged metrics. We discuss some limitations of LCMs in Section 5.1.6.

| Validation | Micro | | | Macro | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| Baseline | 0.930 | 0.928 | 0.929 | **0.882** | 0.827 | 0.854 |
| LCMs | 0.929 | 0.925 | 0.927 | 0.877 | 0.831 | 0.853 |
| +types | **0.933** | **0.930** | **0.931** | 0.875 | **0.842** | **0.858** |
| Test | 0.926 | 0.927 | 0.926 | 0.842 | 0.833 | 0.837 |

Table 5.3. Micro and Macro averaged (P)recision, (R)ecall, and F1 score of event detection on the validation set of CMED. **Baseline**: No levitated markers; solid markers surrounding the medication span only. **LCMs**: Solid markers and max pooled LCMs. **+types**: Typed levitated markers according to the semantic tags of the verbs. The best numbers in each column are in **bold**.

#### Context Classification

We estimate a separate model for each CC dimension, experimenting with the three context pooling functions described in Section 5.1.3. The levitated window size $W$ and Sparsegen-lin $\lambda$ coefficient were tuned on the validation set. We also report performance of a baseline model with no levitated markers, where predictions are made given only the solid markers surrounding the medication span.

Classification results for each model and each context dimension are given in Table 5.4. As already shown for EE in Table 5.3, our baseline is quite strong, outperforming many of our models. Still, with the exception of the Action dimension, LCMs with windowed sparse attention pooling outperform the baseline with solid markers only. However, there is a large performance drop between validation and test sets across dimensions. We discuss this disparity in more detail in Section 5.1.5. While using the rules described in Section 5.1.3 to choose levitated spans outperforms the baseline in some cases (e.g., Temporality and

Negation), its performance is inconsistent. We provide a potential explanation for this in Section 5.1.5.

| Validation | Action | | | Actor | | | Certainty | | | Negation | | | Temporality | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Baseline | 0.837 | 0.786 | **0.811** | 0.766 | 0.716 | 0.740 | 0.686 | 0.673 | 0.679 | 0.781 | 0.657 | 0.713 | 0.792 | 0.773 | 0.783 |
| Maxpool | | | | | | | | | | | | | | | |
| window | 0.834 | 0.785 | 0.809 | 0.751 | 0.713 | 0.731 | 0.679 | 0.668 | 0.673 | 0.730 | 0.677 | 0.703 | **0.801** | 0.797 | 0.799 |
| rules | 0.825 | 0.784 | 0.804 | 0.722 | 0.730 | 0.725 | 0.661 | 0.652 | 0.656 | 0.789 | 0.679 | 0.730 | 0.795 | 0.792 | 0.793 |
| Softmax | | | | | | | | | | | | | | | |
| window | 0.836 | **0.788** | **0.811** | 0.758 | 0.706 | 0.731 | 0.682 | 0.667 | 0.674 | 0.796 | 0.679 | 0.733 | 0.769 | 0.785 | 0.777 |
| rules | **0.838** | 0.781 | 0.809 | 0.745 | **0.732** | 0.738 | 0.682 | 0.661 | 0.671 | 0.766 | 0.632 | 0.693 | 0.783 | 0.780 | 0.782 |
| Sparse | | | | | | | | | | | | | | | |
| window | 0.835 | 0.780 | 0.806 | **0.799** | 0.709 | **0.751** | **0.707** | **0.676** | **0.691** | 0.774 | **0.697** | **0.734** | 0.769 | **0.809** | **0.803** |
| rules | 0.833 | 0.784 | 0.808 | 0.733 | 0.717 | 0.725 | 0.678 | 0.660 | 0.669 | **0.801** | 0.673 | 0.732 | 0.768 | 0.787 | 0.777 |
| Test | Softmax-window | | | Sparse-window | | | Sparse-window | | | Sparse-window | | | Sparse-window | | |
| | 0.881 | 0.744 | 0.793 | 0.559 | 0.549 | 0.554 | 0.583 | 0.626 | 0.591 | 0.491 | 0.500 | 0.495 | 0.590 | 0.592 | 0.591 |

Table 5.4. Macro averaged (P)recision, (R)ecall, and F1 score of context classification on CMED for each pooling method and context dimension. The best score for each metric is in **bold**. The model for each dimension with the best F1 score on the validation set was evaluated on the test set given gold standard event input (Gold) and predictions from the best EE model (E2E). **Baseline**: No levitated markers; solid markers surrounding the event span only. **Maxpool**: Max pooling of the levitated marker representations. **Softmax**: Attention pooling with softmax projection. **Sparse**: Attention pooling with Sparsegen-lin projection.

#### Cue Detection

We performed a manual evaluation of the cues detected by the sparse attention pooling mechanism by inspecting the spans with the highest attention weight. We note that, while softmax pooling outperformed sparse pooling for the Action dimension, softmax tended to produce approximately uniform weights across spans, meaning it was not useful for cue detection.

The author annotated a random, stratified subset of 50 cue predictions on the validation set from each dimension as either "Relevant" or "Irrelevant" to the predicted label. More specifically, an example was marked "Relevant" if any of the spans with a non-zero attention weight were semantically relevant to the prediction made by the model. For example, "not" and "neither" are semantically relevant cues to the Negated label. In the majority of cases, the attention weights assigned by the models were highly sparse, selecting only a small subset of the input tokens. However, in cases where the attention weights were non-sparse we marked an example according to the spans with weights $> 0.1$.[5] Due to the infrequency of "Negated" examples, Negation cue predictions were sourced from both the validation and train sets. We report the percentage of "Relevant" cues.

| Dimension | % Relevant Cues | Dimension | % Relevant Cues |
|---|---|---|---|
| Action | 65 | Negation | 60 |
| Certainty | 44 | Temporality | 56 |

Table 5.5. Accuracy of the cues detected by the sparse attention pooling mechanism, computed as the percentage of Relevant cues.

[5]The mean of the attention weights for the annotated examples fell between 0.1 and 0.2, depending on the model.

We performed this evaluation for each context dimension with the following exceptions: we excluded the Actor dimension as nearly all cues were either implicit or in the document-level context; for Certainty we excluded the "Certain" label, as there are no cues for these examples; likewise, we only evaluated "Negated" instances. Table 5.5 shows the percentage of Relevant cues detected by the sparse attention pooling mechanism for each context dimension. We also provide examples of Relevant and Irrelevant cues detected for each context dimension in Figure 5.3.

We see that the cues detected by the sparse attention mechanism are often relevant to the predicted label, at around 60% for Action, Negation, and Temporality. For Action, the detected cues are relevant verbs such as "increase", "add", or "reduce". For Negation, detected cues include both syntactic (e.g., "not") and lexical (e.g., "hold off", "declines"). For Temporality, the cues include both auxiliary verbs (e.g., "will") and verbs of a relevant tense (e.g., "started" for Past). Certainty is an exception, with only 44% relevant cues. While cues such as "consider" are often correctly identified, others such as "would" and "suspect" can be missed (cf. Figure 5.3d).

| | |
|---|---|
| **Start** | we will add on subcutaneous **enoxaparin** |
| **UniqueDose** | 1 . id . received 1 g **vanc** 1 g cefepime in ed . |
| **Decrease** | will therefore reduce **levothyroxine** to 175 mcg |

(a) Relevant cues for Action.

taking glyburide 5 mg qd instead of **metformin**
**insulin 70 / 30** 80 qam 70 qpm ( recently increased simvastatin
i will change his nph to **lantus** 10units

(b) Irrelevant cues for Action.

consider addition of **abx** for hap
may need higher dose of **acei**
will refer to cardiology regarding whether to **anticoagulate** him .

(c) Relevant cues for Certainty. Labels for all examples are "Hypothetical" or "Conditional".

i suspect we will be able to discontinue the **glyburide** .
would like to change from **prozac** to something else .
**lisinopril** 10 mg daily 5 . pt will likely need fibrate

(d) Irrelevant cues for Certainty. Labels for all examples are "Hypothetical" or "Conditional".

she prescribed an increase in his **atenolol** from 25mg to 50mg . he has not yet completed that prescription .
she refused **nitroglycerine** .
declines **coumadin** .

(e) Relevant cues for Negation. All examples are "Negated".

plan med managnement . **bb** increased today since pt . never increased
one week pta he decided to abruptly discontinue his **fentanyl** patch
would like to start **niacin** but pt not interested at this time .

(f) Irrelevant cues for Negation. All examples are "Negated".

| | |
|---|---|
| **Present** | increase **glipizide** to 5 mg po qd b . |
| **Future** | she will increase her **humulin** |
| **Past** | was recently started **insulin** |

(g) Relevant cues for Temporality.

**glucophage** , to be resumed at 500mg po bid in two days ;
changed to **isordil** . continuing home meds of nifedipine and asa .
he will certainly need an adjustment of his **antihypertensive regimen** .

(h) Irrelevant cues for Temporality.

Figure 5.3. Example cues detected by LCMs with sparse attention pooling on each context dimension. Examples of Relevant cues from each context dimension. The **medication** is in orange and **cues** are highlighted in green proportional to their attention weight. Gold labels are provided where necessary to aid interpretation of the cues.

### 5.1.5 Discussion

As mentioned in Section 5.1.4, there is a large performance drop between validation and test sets for all context dimensions except Action. We here provide an explanation of this

performance difference by way of a brief error analysis.

For Actor, Certainty, and Temporality the performance drop is due almost exclusively to poor performance on the "Unknown" label. This label is very infrequent in the dataset, making it very difficult to learn generalizable representations (cf. Table 5.2 for dataset statistics). For example, there are six "Unknown" Certainty examples in the test set, only two in the train set, and none in the validation set. Additionally, few test examples for a given label means that a single incorrect prediction has a large effect on the macro averaged metrics.

The performance discrepancy for Negation is due to poor test performance on the "Negated" label, which is quite infrequent. A review of Negated instances across the splits revealed a large variety of expressions of negation including lexical (e.g., "Coumadin was *deferred*") and requiring anaphora resolution (e.g., "We dicussed narcotics. We *elected against* these medications."). Few examples and a variety of expressions again means learning generalizable representations is difficult.

### 5.1.6 Limitations and Future Work

This study has the following limitations: The small size of the validation and test sets for the CC subtask means that it is difficult to be certain regarding performance differences between methods. We addressed this by training and evaluating our models on multiple cross-validation splits and with multiple random initializations, but the confidence intervals for each metric are quite wide (around 20% across dimensions). As discussed in Section 5.1.5, the small dataset size also hinders the models' ability to generalize to the test examples. Future work should therefore expand the amount of data available for both training and evaluation, especially regarding the less frequent labels.

We also found that our rule-based levitated markers could be too limiting. For example, while marking verbs our model missed clinical shorthand cues such as "d/c" ("discontinued") and our negation cue list was missing some lexical cues (e.g., "hold", "deny") that were prevalent in the data. Missed verbs were therefore excluded from the subsequent semantic tagging for the EE task and were never assigned the relevant type. Future work will be to develop a hybrid approach in which rules guide the model without overly restricting it.

### 5.1.7 Implementation Details

All models were implemented using PyTorch [233]. Model training and evaluation was performed on a NVIDIA Tesla A100 with 40GB of VRAM. Hyper-parameter settings for each model are detailed in tables 5.6 and 5.7.

| | |
|---|---|
| PLM | BlueBERT |
| Max sequence length | 300 |
| Batch size | 32 |
| Epochs | 10 |
| Learning rate | 4e-5 |
| Auxiliary tasks | Action |
| Pooling method | Max pooling |

Table 5.6. Hyperparameter settings for the Event Extraction models.

| | Action | Actor | Certainty | Negation | Temporality |
|---|---|---|---|---|---|
| PLM | BlueBERT | BERT-base | ClinicalBERT | ClinicalBERT | ClinicalBERT |
| Max sequence length | 300 | 300 | 300 | 300 | 300 |
| Batch size | 32 | 32 | 32 | 32 | 32 |
| Epochs | 20 | 10 | 20 | 20 | 20 |
| Learning rate | 3e-5 | 3e-5 | 3e-5 | 3e-5 | 3e-5 |
| Auxiliary data | i2b2 2009 | - | i2b2 2009 | i2b2 2009 | i2b2 2009 |
| Auxiliary tasks | - | - | - | Action | - |
| $W$ | 10 | 5 | 10 | 10 | 5 |
| $\lambda$ | 0.5 | 0.5 | 0.5 | 0.2 | 0.5 |

Table 5.7. Hyperparameter settings for the Context Classification models. $W$: window size parameter for the Levitated Context Markers. $\lambda$: sparsity coefficient for the Sparsegen-lin projection function.

## 5.2 Levitated Context Markers for Multi-Task FEE

The previous experiments show that LCMs not only outperform a strong baseline model on CMED, but also are able to detect relevant cue spans –e.g., negation and speculation cues, and verbs of change– in an unsupervised fashion, adding interpretability to the model predictions. However, a clear limitation of this research is it's pipeline approach, which treats each context dimension as a separate task requiring its own model. Chapter 4 demonstrated that event type, polarity, and uncertainty are independent and that this independence can be mirrored in a multi-task prediction model without additional objectives besides that coming from the task signals. More practically, we can estimate a single multi-task FEE model by simply adding task-specific heads to a generic input encoder. FactVAE from Chapter 4, for example, modeled each task using linear layers given the final hidden state from a transformer encoder. This section extends the LCM model from Part 1 to a true multi-task approach which targets FEE prediction performance on SemMed Context. Additionally, where the previous approach was evaluated only on very simple, single-argument events, the extension is evaluated on SemMed Context triples.

### 5.2.1 Extended Model

The first step is to generalize the model given in Equation (5.1) to handle more complex event structures. We modify the notation in Equation (5.1) slightly to incorporate a hidden representation of the event $h_e$, computed using a generic event encoder function $\phi$ from the $N$ event arguments $x_{argN}$. In our previous CMED experiments where $N = 1$, $\phi$ is simply the identity function, but for more complex event schemas $\phi$ could incorporate more advanced functionality such as the role layer from [106] which enumerates pairs of potential arguments and assigns a role to each.

$$h_e = \phi(PLM(x_{arg1}, ..., x_{argN})) \tag{5.9}$$

$$\hat{y} = f([h_e; c(h_\ell)]) \tag{5.10}$$

Given the hidden event representation $h_e$, we then compute attention between it and the levitated context representations $c(h_\ell)$ as previously described in Section 5.1.3.

We experiment with LCMs on SemMed Context which represents events as (subject, predicate, object) triples. This means that events have two arguments ($N = 2$). We define the event encoding function $\phi$ as the concatenation of the hidden representations of the subject and object spans, i.e., $h_e = [h_s; h_o]$. We did perform preliminary experiments which projected the concatenated subject and object representations into a single, lower-dimensional representation (i.e., $h_e = \text{Linear}([h_s; h_o])$), but these generally performed worse than concatenation.

As for CMED, we experiment with 5 model variants on SemMed Context:

- **CLS**: A linear layer predicts the target label from the pooled output of a pretrained encoder.
- **Solid**: The model described in Equation (5.9) but without the context function $c()$, i.e., $\hat{y} = f(h_e)$.
- **Levitated**: We use max pooling as the context function (cf. Equation (5.4)).
- **Softmax**: We use softmax pooling as the context function (cf. Equation (5.5)).
- **Sparse**: We use sparsegen-lin as the context function (cf. Equation (5.8)).

| | Negation | | | | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|
| | Negative | | | Positive | | | Macro Avg. | | |
| | P | R | F1 | P | R | F1 | P | R | F1 |
| CLS | 0.479 | 0.492 | 0.485 | 0.995 | 0.995 | 0.995 | 0.792 | 0.781 | 0.785 |
| Solid | 0.658 | 0.548 | 0.598 | 0.995 | 0.995 | 0.995 | 0.955 | 0.952 | 0.953 |
| Levitated | 0.653 | 0.623 | **0.638** | 0.996 | 0.997 | 0.996 | 0.952 | 0.955 | 0.951 |
| + Softmax | 0.667 | 0.577 | 0.619 | 0.996 | 0.997 | 0.996 | 0.954 | 0.953 | 0.953 |
| + Sparse | 0.754 | 0.531 | 0.623 | 0.995 | 0.998 | 0.997 | 0.952 | 0.956 | **0.954** |

| | Certainty | | | | | |
|---|---|---|---|---|---|---|
| | Uncertain | | | Certain | | |
| | P | R | F1 | P | R | F1 |
| CLS | 0.719 | 0.818 | 0.765 | 0.979 | 0.963 | 0.971 |
| Solid | 0.883 | 0.869 | 0.876 | 0.985 | 0.987 | 0.986 |
| Levitated | 0.892 | 0.863 | 0.877 | 0.984 | 0.988 | 0.986 |
| + Softmax | 0.883 | 0.869 | 0.876 | 0.985 | 0.987 | 0.986 |
| + Sparse | 0.877 | 0.884 | **0.880** | 0.987 | 0.986 | 0.986 |

Table 5.8. (P)recision, (R)ecall, and F1 score for the single-task models on the Polarity, Certainty, and Predicate tasks, respectively. The best F1 score for each task is in **bold**. N.B., the results for the CLS model are the same as those reported in Table 3.9 in Chapter 3.

### 5.2.2 Preliminary Experiments

To establish a baseline for comparison, we first perform a set of single-task experiments in which we estimate a separate model for each of the three tasks: polarity, uncertainty, and predicate type. We estimate each model and tune hyperparameters on the train and validation splits of SemMed Context, respectively, as described in Table 3.8. We train for 5 epochs with a batch size of 32 and a learning rate of 2e-5. For the Sparse model we set $\lambda = 0.5$. The precision, recall, and F1 score of each model on the validation split are given in Table 5.8.

We then estimated multi-task models which jointly predict polarity, uncertainty, and predicate type by stacking three prediction heads on top of the BERT encoder. Besides this slight architecture change, these models have identical hyperparameters to their single-task counterparts. The performance of these models on the validation set is given in Table 5.9.

| | Polarity | | | | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|
| | Negative | | | Positive | | | Macro Avg. | | |
| | P | R | F | P | R | F | P | R | F |
| CLS | 0.552 | 0.489 | 0.518 | 0.995 | 0.996 | 0.995 | 0.788 | 0.780 | 0.783 |
| Solid | 0.667 | 0.601 | 0.637 | 0.996 | 0.997 | 0.997 | 0.951 | 0.951 | 0.951 |
| Levitated | 0.492 | 0.748 | 0.594 | 0.997 | 0.992 | 0.995 | 0.948 | 0.950 | 0.949 |
| + Softmax | 0.632 | 0.659 | **0.645** | 0.996 | 0.997 | 0.997 | 0.952 | 0.949 | 0.951 |
| + Sparse | 0.624 | 0.620 | 0.622 | 0.995 | 0.996 | 0.996 | 0.952 | 0.951 | 0.952 |

| | Uncertainty | | | | | |
|---|---|---|---|---|---|---|
| | Uncertain | | | Certain | | |
| | P | R | F | P | R | F |
| CLS | 0.774 | 0.773 | 0.774 | 0.974 | 0.974 | 0.974 |
| Solid | 0.870 | 0.891 | **0.881** | 0.987 | 0.985 | 0.986 |
| Levitated | 0.856 | 0.890 | 0.873 | 0.987 | 0.983 | 0.985 |
| + Softmax | 0.885 | 0.860 | 0.872 | 0.984 | 0.987 | 0.986 |
| + Sparse | 0.869 | 0.890 | 0.879 | 0.987 | 0.985 | 0.986 |

Table 5.9. Results for the multi-task models jointly trained for 5 epochs on the Polarity, Certainty, and Predicate tasks. F1 scores higher than the best single-task model from Table 5.8 are in **bold**.

These tables show that all the models outperform the CLS baseline in both the single- and multi-task setups. This is especially true for the negative and uncertain labels where we see improvements greater than 0.1 F1 over the baseline for both single- and multi-task models. Further, we see very little difference in performance among the non-baseline models and between the single- and multi-task models, indicating that the BERT encoder is able to learn a single set of token representations that are informative for all tasks.

#### Cue Detection

We here compare the Sparse models from the single- and multi-task experiments on cue detection. As in Section 5.1, we manually annotated 50 examples predicted by the Sparse model for each negative and uncertain examples, and 10 examples of each predicate label as either relevant or irrelevant. We then and computed the percentage of relevant cues for the single- and multi-task setups. We annotated the same set of examples for each model for fair comparison. To investigate the sparsity of the attention mechanism, we also computed token

coverage as the percentage of tokens receiving a non-zero attention weight. The results are given in Table 5.10.

| | % of Relevant Cues | | | % token coverage | | |
|---|---|---|---|---|---|---|
| | Negative | Uncertain | Predicate | Negative | Uncertain | Predicate |
| Single-Task | 92 | 73 | 52 | 10 | 15 | 30 |
| Multi-Task | 53 | 73 | 48 | 28 | 28 | 28 |

Table 5.10. Percentage of correct predictions with relevant cues from the sparse LCM model on the validation set.

While the single-task models had very similar prediction performance to the multi-task models, they performed better in the cue detection task –most notably on negative examples. We suspect that the drop in performance on polarity and predicate type, compared to the similar performance on uncertainty, is a result of the model "choosing" which task to focus its internal representation on. We also note that the somewhat poor cue detection performance of the multi-task models as well as the single-task predicate model included more uniform weights as measured by the token coverage. Furthermore, even though the single- and multi-task models performed similarly on uncertainty, the higher coverage of the multi-task model means more non-relevant tokens are selected in addition to the relevant cues, potentially reducing interpretability of the model predictions. Anecdotally from our manual review, we note that token coverage of 10-15% was suitable for interpretability while higher coverage models tended to assign small attention weights to many irrelevant spans.

### 5.2.3 Improving Multi-Task Performance on Cue Detection

Given the findings above, we aim to improve the multi-task Sparse model's performance on cue detection without sacrificing prediction performance. We develop model variants based on two hypotheses regarding the failure of the multi-task model on cue detection.

**Hypothesis 1:** In the single-task models the BERT encoder is able to specialize, learning hidden token representations that are task-oriented which result in the detection of correlations between semantically relevant cues and the target task. In the multi-task model, however, the encoder must learn hidden token representations that are applicable to all tasks. While the experiments above demonstrate that the encoder is able to do this without hurting prediction performance, it may reduce the correlations between hidden representations and each task. To encourage the model to learn task-specific token representations we add a single task-specific layer for each task after the shared PLM encoder. Modifying the model above, we introduce as task-specific layer $\tau$.

$$h_e = \tau([h_s; h_o]) \tag{5.11}$$

$$\hat{y} = f([h_e; c(h_\ell)]) \tag{5.12}$$

We experiment with two definitions of $\tau$: a BERT layer and a projected attention layer (PAL). The BERT layer is simply another instance of multi-headed attention followed by layer normalization and a feed forward layer. The PAL was introduced in [234] as a more parameter-efficient alternative to additional BERT layers. The PAL is a multi-headed attention mechanism sandwiched between an encoder and decoder. The encoder and decoder allow us to project the input (i.e., the token representations output by BERT) into a smaller space and thus reducing the total number of parameters used. Formally, PAL is defined as

$$\text{PAL}(\boldsymbol{h}) = V^D \text{Attn}(V^E \boldsymbol{h}) \tag{5.13}$$

where $V^E$ and $V^D$ and the encoder and decoder matrices, respectively, Attn() is multi-headed attention, and $\boldsymbol{h}$ denotes the token representations. In practice, adding a BERT layer for each task requires an additional 21 million parameters, while a PAL for each task requires only an additional 4 million due to the dimensionality reduction.

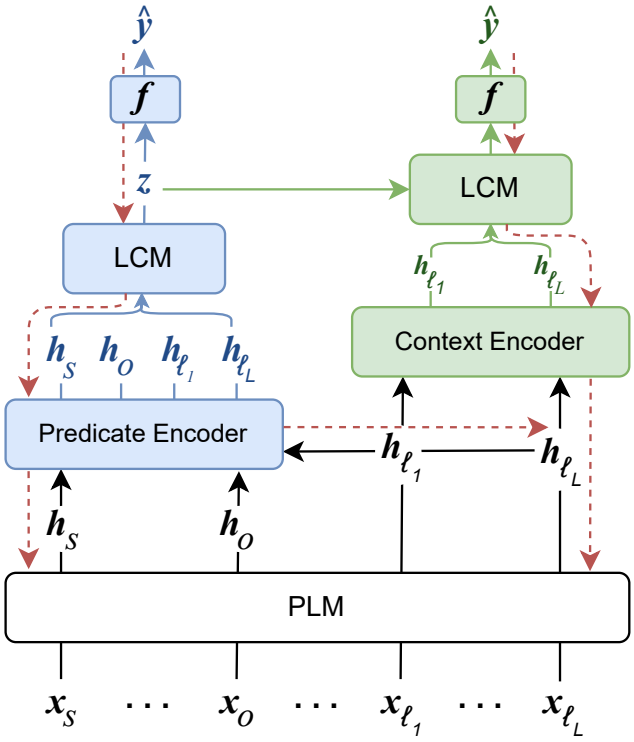



Figure 5.4. An illustration of the hierarchical levitated context marker model. A shared pretrained language model (PLM) encodes the input tokens $\boldsymbol{x}$ into hidden representations $\boldsymbol{h}$. Then, task-specific layers further encode the hidden representations and pass them to levitated context marker (LCM) layers. In this hierarchical model, the output $\boldsymbol{z}$ of the LCM layer for the predicate prediction task (in **blue**) is input to the LCM layer for the auxiliary tasks (in **green**). The **red** dashed arrows indicate the path of gradient back-propagation. Note that gradients from the auxiliary tasks are not propagated through $\boldsymbol{z}$ into the layers meant for the predicate task, in order to keep the task-specific layers independent.

**Hypothesis 2:** The models above treated each task –polarity, uncertainty, and predicate type– as entirely independent. While Chapter 4 discussed how these dimensions are indeed semantically independent (we can change the value of each without affecting the others), we note that practically in a multi-task FEE model we can only reliably determine the polarity and

uncertainty status once we know what the predicate is. Therefore, we develop a hierarchical model which first predicts the predicate and then the polarity and uncertainty based on the predicate representation. Building off the task-specific models from Hypothesis 1, we first compute a latent event encoding $z$ from the predicate task encoder and an LCM layer. We then use $z$ as input to the polarity and uncertainty LCM layers. This model is depicted graphically in Figure 5.4 and associated equations are given in Equation (5.14), colored to match the components in the figure.

$$\begin{aligned} \boldsymbol{h}_e &= \tau([\boldsymbol{h}_s; \boldsymbol{h}_o]) \\ \boldsymbol{h}_\ell &= \tau(\boldsymbol{h}_\ell) \end{aligned} \qquad \begin{aligned} \boldsymbol{z} &= [\boldsymbol{h}_e; c(\boldsymbol{h}_\ell)] \\ \hat{y} &= f(\boldsymbol{z}) \end{aligned} \qquad \begin{aligned} \boldsymbol{h}_\ell &= \tau(\boldsymbol{h}_\ell) \\ \boldsymbol{z} &= [\boldsymbol{z}; c(\boldsymbol{h}_\ell)] \\ \hat{y} &= f^{(c)}(\boldsymbol{z}) \end{aligned} \tag{5.14}$$

where $\tau$ is the task-specific layer (BERT or PAL).

Importantly, as depicted in Figure 5.4, we do not back-propagate the training signal from the context tasks $\hat{y}$ to through the hidden representation of the predicate $z$. This keeps the predicate-specific layers independent from the polarity and uncertainty layers, while still allowing these context tasks to utilize the predicate representation.

#### Results

As before we evaluate each model on the SemMed Context validation set. We keep hyperparameters the same (cf. Section 5.2.2) and set the dimensionality of the PALs to 204 following [234]. Table 5.11 shows the predictions results for the flat and hierarachical sparse models with both BERT and PAL task layers compared to the model without any task-specific layers.

| | | Polarity | | | | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Negative | | | Positive | | | Macro Avg. | | |
| Model | Task Encoder | P | R | F | P | R | F | P | R | F |
| Flat | None | 0.624 | 0.620 | 0.622 | 0.995 | 0.996 | 0.996 | 0.952 | 0.951 | **0.952** |
| | Bert Layer | 0.583 | 0.679 | 0.627 | 0.997 | 0.995 | 0.996 | 0.947 | 0.949 | 0.948 |
| | PAL | 0.605 | 0.669 | 0.636 | 0.997 | 0.996 | 0.996 | 0.948 | 0.949 | 0.949 |
| Hier | Bert Layer | 0.619 | 0.672 | **0.645** | 0.997 | 0.996 | 0.996 | 0.948 | 0.950 | 0.949 |
| | PAL | 0.652 | 0.633 | 0.642 | 0.996 | 0.997 | 0.996 | 0.949 | 0.946 | 0.947 |

| | | Uncertainty | | | | | |
|---|---|---|---|---|---|---|---|
| | | Uncertain | | | Certain | | |
| | | P | R | F | P | R | F |
| Flat | None | 0.869 | 0.890 | **0.879** | 0.987 | 0.985 | 0.986 |
| | Bert Layer | 0.857 | 0.890 | 0.873 | 0.987 | 0.983 | 0.985 |
| | PAL | 0.870 | 0.880 | 0.875 | 0.986 | 0.985 | 0.985 |
| Hier | Bert Layer | 0.888 | 0.862 | 0.875 | 0.984 | 0.988 | 0.986 |
| | PAL | 0.850 | 0.897 | 0.873 | 0.988 | 0.982 | 0.985 |

Table 5.11. Prediction performance for the flat vs hierarchical sparse LCM models jointly trained on the polarity, uncertainty, and predicate type tasks.

Compared to the flat model without task layers, we see that both the BERT layer and PAL improve prediction performance on the negative label by up to $0.014$, but slightly reduce it

on the uncertain label ($-0.006$) and predicate macro F1 ($-0.003$). Comparing the flat and hierarchical models, we see further improvements on the negative label (up to $0.645$ F1) and very similar performances on uncertainty and predicate.

To test our hypotheses regarding the failure of the multi-task LCM models on cue detection, we evaluate each of the above models on the same set of examples as before. Table 5.12 shows the results of this evaluation compared to the previous results from the single- and multi-task models.

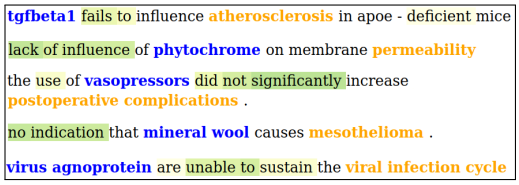
| | | % of Relevant Cues | | | | % Token Coverage | | |
|---|---|---|---|---|---|---|---|---|
| Model | Task Encoder | Negative | Uncertain | Predicate | Avg. | Negative | Uncertain | Predicate |
| Single-task | | **92** | 73 | 52 | 72 | 10 | 15 | 30 |
| Flat | None | 53 | 73 | 48 | 58 | 28 | 28 | 28 |
| Flat | BertLayer | 74 | 76 | 32 | 61 | 13 | 13 | 10 |
| Flat | PAL | 70 | 60 | 51 | 60 | 25 | 26 | 21 |
| Hier | BertLayer | 40 | 50 | 17 | 36 | 13 | 11 | 11 |
| Hier | PAL | 84 | **90** | **56** | **77** | 15 | 19 | 13 |

Table 5.12. Results of the sparse attention models on the cue detection task, including the percentage of predictions with relevant cues as well as average input token coverage of the sparse attention masks. Flat-ST refers to the single-task models with sparse attention from Table 5.8.

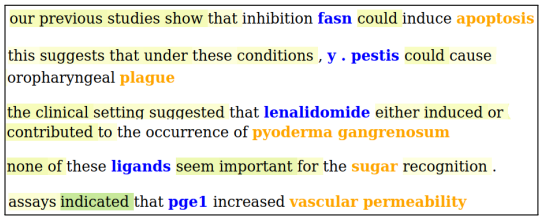
Regarding Hypothesis 1 –that task-specific layers would improve performance on cue detection– the results are somewhat supportive for the Flat models, with both BERT layer and PAL models obtaining a higher average % of relevant cues than the model without task layers. Additionally, both models reduce the token coverage by more than half, which as discussed above is generally better for interpretability.

Regarding Hypothesis 2 –that a hierarchical model would be more informative– the results are inconclusive. The hierarchical model with the BERT layer performs poorly on cue detection with an average relevance of only 36%, but the PAL model achieves the highest average relevance at 77% and achieves the highest per-task relevance for uncertainty and predicate as well as second highest for polarity. We offer the following hypothesis for this discrepancy. The BERT layer is significantly larger than the PAL, with 7.1 million and 1.3 million parameters, respectively. As such it is also more powerful and able to transform the token representations output by the BERT encoder into representations that may be useful for prediction (i.e., they are linearly separable) but are more semantically distant from the actual meaning of the words. This is supported by the low token coverage of the BERT layer model (13%, 11%, and 11%) as it is able to achieve respectable prediction performance from very few semantically irrelevant tokens. In contrast, the more limited capacity of the PAL model is unable to make such drastic transformations to the encoder token representations, forcing the model to find semantically relevant correlations between tokens and the target task. However, these conjectures would need to be supported with experimental evidence in future work.

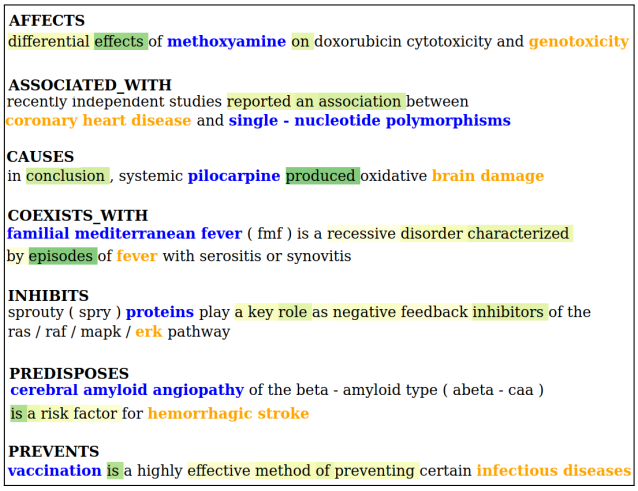
To illustrate the types of cues detected by the hierarchical projected attention model, we collected examples of relevant cues on the validation set. These are given in Figure 5.5. To further demonstrate the potential interpretability of the model outputs, Figure 5.6 shows the cues detected from a single PREVENTS claim that is both negative polarity and uncertain.




(a) Detected negation cues.




(b) Detected uncertainty cues.




(c) Detected predicate cues.

Figure 5.5. Sparse attentions from the Hierarchical Projected Attention model on the polarity, uncertainty, and predicate tasks showing the detected cues. **Subject** entities are in blue, **objects** in orange, and **cue spans** are highlighted in green proportional to their attention weight.

In Figure 5.5 we see that across the polarity, uncertainty, and predicate tasks the model detects a variety of relevant spans. These include not only common cue words such as "*not*" and "*seem*" but also cue phrases including lexical markers of negation and speculation such as "*lack of*" and "*this suggests that*". Also on the predicate task the model is able to find both relevant cue words such as "*effects*" (AFFECTS) and "*produced*" (CAUSES) and phrases such as "*characterized by*" (COEXISTS_WITH) and "*is a risk factor*" (PREDISPOSES).

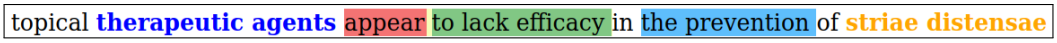



Figure 5.6. A negated and uncertain PREVENTS claim highlighted with the highest attention-weighted cue spans for each context dimension, colored as follows: **polarity**, **uncertainty**, **predicate**. The **subject** entity is blue and the **object** is orange.

### 5.2.4 Results on Test Splits

From the results above, the hierarchical PAL model is deemed the overall best performing with the highest average cue relevance and respectable prediction performance. We therefore evaluate this model on both the SemMed Context and SemRep Factuality test splits. These results are given in Table 5.13 and compared to the baseline results from Section 3.3.

We see large increases in performance over the baseline on both SemMed Context and SemRep Fact test splits. Notably on SemRep Fact the F1 score doubles for the negative label (0.243 $\rightarrow$ 0.488) and nearly doubles for the uncertain label (0.430 $\rightarrow$ 0.753). There is also an 0.28 increase in the macro F1 score on the predicate task. Compared to the performance of the rule-based SemRep Factuality model evaluated in Table 3.2, our model performs better on the Uncertain label (0.753 vs. 0.690), the same on the positive label, and only slightly worse on the certain label (0.955 vs 0.975). Our model does, however, perform worse on the negative label (0.488 F1 vs. 0.698).

We also manually evaluate the cues detected by our model on both test splits. As before,

| | | Polarity | | | | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Negative | | | Positive | | | Macro Avg. | | |
| | Model | P | R | F | P | R | F | P | R | F |
| SemMed Context | Baseline | 0.462 | 0.517 | 0.488 | 0.995 | 0.994 | 0.995 | 0.795 | 0.778 | 0.785 |
| | Hier PAL | 0.621 | 0.651 | 0.636 | 0.997 | 0.996 | 0.996 | 0.950 | 0.950 | 0.950 |
| SemRep Fact | Baseline | 0.273 | 0.220 | 0.243 | 0.982 | 0.997 | 0.989 | 0.700 | 0.619 | 0.637 |
| | Hier PAL | 1.000 | 0.323 | 0.488 | 0.976 | 1.000 | 0.988 | 0.914 | 0.924 | 0.917 |

| | | Uncertainty | | | | | |
|---|---|---|---|---|---|---|---|
| | | Uncertain | | | Certain | | |
| | Model | P | R | F | P | R | F |
| SemMed Context | Baseline | 0.720 | 0.821 | 0.767 | 0.979 | 0.962 | 0.970 |
| | Hier PAL | 0.853 | 0.907 | 0.879 | 0.989 | 0.982 | 0.985 |
| SemRep Fact | Baseline | 0.315 | 0.677 | 0.430 | 0.977 | 0.904 | 0.939 |
| | Hier PAL | 0.694 | 0.823 | 0.753 | 0.970 | 0.940 | 0.955 |

Table 5.13. Results of the baseline BERT model on both SemMed Context and the SemRep Factuality test set. For predicate the model was evaluated on the subset of the SemRep Factuality test set that contained the predicates used in SemMed Context.

we randomly select 50 negative and 50 uncertain examples as well as 10 examples from each Predicate type. The results are given in Table 5.14, which shows that the ability of the hierarchical PAL model to identify relevant cues generalizes to both corpora, achieving even higher average Relevance than on the validation set.

| | % of Relevant Cues | | | | % Token Coverage | | |
|---|---|---|---|---|---|---|---|
| Test Set | Negative | Uncertain | Predicate | Avg. | Negative | Uncertain | Predicate |
| SemMed Context | 96 | 86 | 65 | 82 | 15 | 19 | 14 |
| SemRep Fact | 90 | 88 | 59 | 79 | 15 | 20 | 14 |

Table 5.14. Cue detection results of the hierarchical projected attention model on the SemRep Context and SemRep Factuality test splits.

## 5.3 Conclusion

This chapter introduced Levitated Context Markers (LCMs), a novel method for predicting information regarding events. We first established the usefulness of LCMs on CMED, a small clinical dataset for EE and CC annotated with 5 context dimensions. These experiments demonstrated that LCMs not only outperform a strong baseline model, but also that their sparse attention mechanism is able to automatically identify relevant cue tokens in the input. Going further, we modified LCMs to allow for a generalized event representation and evaluated them on SemMed Context, achieving good performance on multi-task FEE without sacrificing cue detection ability.

While the generalized event representation was shown to perform well on simple triples, the practicalities of applying it to more complex event representations remains a problem for future work. For example, events in GENIA are built up from potentially non-contiguous terms, can have multiple themes or causes, and can also be nested, such that the theme or cause of one event is an event itself [235]. The LCMs described so far have used solid markup tokens to identify the terms in an event, but doing so for a corpus like GENIA may

require too many markup tokens to be feasible.

Additionally, while we provide a simple evaluation of the interpretability aspect of LCMs in Section 5.1.4 and Section 5.2.4, a more in-depth evaluation is needed. Because LCMs can be categorized as an explainability technique, they ought to be evaluated on other tasks and compared to other methods under XAI such as gradient-based (e.g., integrated gradients [236], [237]), perturbation-based (e.g., token perturbation [238]), and surrogate model methods (e.g., LIME [239] and SHAP [240]) using standard evaluation metrics like sufficiency and comprehensiveness [241].

# Chapter 6

# Conclusion

This thesis set out to develop methods for factualized event extraction (FEE) from the biomedical literature. Specifically, we set out the following researching aims:

- **RA1:** To theoretically and practically demonstrate the importance of treating polarity and uncertainty –and therefore factuality– as central dimensions of the event representation.
- **RA2:** To develop novel machine learning models for FEE that not only achieve good prediction performance but also learn representations of factuality that align with the definition of factuality given in Definition 1.

Towards RA1, Section 1.1 provided a theoretical justification for the integration of factuality into event representations and showed how factuality can be decomposed into dimensions of polarity and uncertainty. Experiments in Chapter 4 demonstrated that while polarity and uncertainty work together to comprise factuality, they are orthogonal dimensions and that it is possible to learn internal model representations that reflect this independence. Section 3.4 described a proof-of-concept biomedical knowledge graph in which each edge contained a factuality representation, illustrating the practical benefit to including factuality in the event representation.

Towards RA2, we experimented with two model types. The BART-based VAEs in Section 4.7 achieved good prediction performance while also maintaining disentanglement of the polarity, uncertainty, and predicate representations. One model variant, FactVAE, used a reparameterized Dirichlet distribution to model the latent factuality variable, which aligns precisely with Definition 1. The LCM models in Chapter 5 improved over FactVAE in prediction performance while also providing interpretable predictions for each task by learning to highlight relevant subspans in the input.

## 6.1 Summary of Results

Table 6.1 gives the performance of each type of model (baseline, VAE, and LCM) on the test split of SemMed Context. We report the results of FactVAE as it outperformed SubjVAE in

Chapter 4 and the hierarchical LCM model with PALs as the best performing model variant from Chapter 5.

The baseline method –single-task fine-tuned BERT models each with a linear prediction layer– achieved respectable performance across tasks, although we note that they performed significantly worse on the less frequent labels in the Negation and Certainty tasks (cf. Table 3.9). Both FactVAE and the LCM model improved over this baseline considerably, with an increase of around 0.05 macro F1 on polarity and uncertainty and 0.15 on the predicate task. Impressively, both of these models utilize a single PLM of a similar size to each of the single-task baseline BERT models, meaning that the performance increase is attributable to our modeling improvements (cf. Table 5.9 for a direct comparison of the LCM model to the baseline).

| | Polarity | | | Uncertainty | | | Predicate | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Baseline | 0.729 | 0.756 | 0.742 | 0.850 | 0.892 | 0.869 | 0.795 | 0.778 | 0.785 |
| FactVAE | 0.769 | 0.859 | 0.807 | 0.940 | 0.937 | 0.938 | 0.934 | 0.931 | 0.932 |
| LCM Hier PAL | 0.809 | 0.824 | 0.816 | 0.921 | 0.945 | 0.932 | 0.950 | 0.950 | 0.950 |

Table 6.1. Macro-averaged results of the baseline BERT model, FactVAE, and the best LCM model variant (Hierarchical with PAL layers) on the SemMed Context test split.

Besides obtaining good prediction performance, FactVAE utilized a single latent distribution to model both polarity and uncertainty that was based on Subjective Logic (SL). This latent distribution was able to independently model each dimension, while also providing an interpretable representation of event factuality as a coordinate on the simplex. We also connected the use of this latent distribution to recent work on data perspectivism and learning from disagreements, showing how –once data becomes available– FactVAE's latent Dirichlet distribution is a natural choice for representing textual uncertainty, a task with high annotator disagreement.

LCMs not only outperformed FactVAE on the polarity and predicate tasks but were also able to automatically identify subspans of the input text that are semantically relevant to the target label across tasks, lending interpretability to the model's predictions. Our evaluation showed that the model highlighted relevant subspans for an average of 82% of examples across tasks (cf. Table 5.14). This ability is, furthermore, a side-effect of the architecture and does not require any additional loss functions, meaning it adds little complexity to model estimation.

## 6.2 Future Work

The research completed for this thesis is only the beginning of what might be explored in relation to both the FEE task and the models developed herein. In this section we will touch upon the limitations of our work and discuss how these might be addressed in the future.

### 6.2.1 Data

SemMed Context is currently the largest dataset of factualized events, and because it is automatically annotated it can feasibly be expanded to cover the entirety of MEDLINE and include all of the predication types in the UMLS Semantic Network. However, automatic annotation means that errors are inevitable. While we attempted to address this by post-hoc label filtering, future work could improve the label quality by soliciting human annotations and using methods such as active learning and programmatic weak supervision to strike a balance between quantity and quality of the annotations. Additionally, the practical applicability of models trained on SemMed Context would be improved if it contained examples of unrelated entity pairs, i.e. subject-object entity pairs that do not belong to any predication, in order guard against false positives, and improved sampling methods could increase the frequency of negative and uncertain examples to create a more balanced dataset.

As mentioned above, using Dirichlet distributions to represent factuality also provides an avenue to leverage crowd annotations. This would be beneficial for the uncertainty dimension, as it has been shown to have high annotator disagreement due to differing subjective perceptions of the degree of uncertainty expressed [133]. Ideally, future work would solicit multiple annotations for each example in order to infer a distributions over more fine-grained labels between "certain" and "uncertain", rather than the binary labels used in this and previous work.

### 6.2.2 Models

On the modeling side, an obvious next step is to combine FactVAE from Chapter 4 and the hierarchical LCM model from Chapter 5 into a model that is able to estimate SL distributions representing event factuality from interpretable subspans of the input. Such a model would be estimated using a variant of the objective function in Equation (4.10) which balances prediction loss and the KL divergence loss of the SL distributions. Once estimated, the model could be applied to large amounts of biomedical text data to construct a knowledge graph similar to the one proposed in Section 3.4. The factuality metadata for each edge in this graph would allow users to both decompose the factuality representations into their constituent claims and inspect the model's prediction for each claim via the detected cue spans. The use of SL distributions would further allow simple post-hoc updates or corrections to the knowledge graph. That is, if a user finds a that the model incorrectly extracted an evidence claim or incorrectly predicted its factuality value, the evidence could be excluded or the SL distribution manually corrected to update the overall claim using SL operations only.

#### VAEs

Computational resource limitations precluded experimenting with baseline BART-based VAE models in Section 4.7, but such experimentation is crucial for understanding the performance

of SubjVAE and FactVAE. To put these models in context it would be helpful to perform an in-depth comparison of their prediction performance, text generation, and disentanglement ability against a standard BART model as well as a BART VAE using Gaussian latents. While the results from the LSTM VAE on general domain data suggested that auxiliary disentanglement objectives are not essential, it would be interesting to explore these objectives in the BART VAEs on SemMed Context.

#### Levitated Context Markers

The LCM models outperformed FactVAE and the BERT baseline on SemMed Context, but it is important to note that there is still significant room for improvement across tasks. Crucially, performance on the negative label is only 0.636 F1 (cf. Table 5.13). The uncertain label too has room for improvement, although it performs considerably better at 0.876 F1. While the macro average F1 on the predicate task is quite high at 0.950, we remind the reader that the version of SemMed Context constructed for this thesis contains a subset of 13 predicate types related to clinical medicine and substance interactions. Future work should evaluate these models on the full set of 30 predicates.

First, we ought to experiment with larger PLMs such as BERT large, which has 400 million parameters compared to BERT base's 140 million and may generalize better to infrequent labels. We should also explore other more modern encoder-decoder PLMs such as T5 [159] and Aya [242] which are pretrained on both free text and a variety of NLP tasks, increasing their generalizability as well as zero- and few-shot performance. Second, while our experiments on CMED showed that using rules to choose the levitated context tokens was too restrictive, it might still be helpful to guide the attention mechanism towards certain cue words (e.g., negation and speculation cues), perhaps by upweighting the attention scores for relevant tokens. However, additional work is required to define precisely how this should be implemented.

On the evaluation side, the experiments in Section 4.7.3 showed that the spans highlighted by the LCM model are relevant to the predictions. However, the sparse attention mechanism is not a deterministic process and it remains to be seen how consistent these spans are across hyper-parameters and datasets. The effect of hyper-parameters could be evaluated by training multiple models using different hyper-parameter values on the same dataset and comparing their predicted attention weights via a vector similarity metric. For evaluating the effect of different training datasets it would be necessary to re-evaluate the relevancy of the output spans to the predicted label. Towards this, it would be helpful to develop a gold-standard dataset of relevant cue spans for each task and label so that evaluation could be computed automatically rather than the ad-hoc manual evaluation we performed here. Besides this, the explainability metrics of sufficiency and comprehensiveness offer the possibility of measuring the relevance of highlighted spans without need for gold-standard annotations [241].

Another practical concern is the applicability of the LCM model to datasets with more com-

plex event annotations. For example, GENIA [16] events are significantly more complex than the SemMed Context triples, as they are composed of more subspans and potentially contain nested events. Using levitated markers may add too many additional tokens to the input. If this is indeed the case (and we suspect it is) how can we get around it? We hypothesized in Chapter 5 that the levitated markers allowed the PLM to learn two representations of the event subspans: a standard in-sentence representation and a separate task-specific representation. It may be possible to still learn both representations without the need for additional marker tokens by using a projected attention layer (PAL) with sparse attention alongside each BERT layer in the PLM, as originally proposed by [234], who themselves hint that "the attention mechanism of PALs can potentially be inspected to see which tokens in a sequence the task-specific parts of the model focus on".

## 6.3 Large Language Models

Technology moves quickly, and in the 3 years since the research for this thesis began we have seen incredible advancement with the release of very large language models (LLMs) such as the GPT [243], [244] and Llama series [245]. These models are being marketed and used as stores of knowledge: they answer questions, guide decisions, and –perhaps most famously– perform programming tasks. It is beyond doubt that these models are able to communicate factual knowledge about the world [246]. However, they are also known to hallucinate knowledge and evidence [247], which has led to their ban on question answering sites such as Stack Overflow [248] and being deemed unsuitable for knowledge-intensive applications such as fact-checking [249]. In other words, these models do not seem to have any real notion of factuality, although this remains to be tested in a controlled setting.

This is even more striking in light of recent work which explores the ability of LLMs to perform latent reasoning over atomic knowledge claims [250]. The extent to which they are able to do so is encouraging, but surely for knowledge based applications such as decision support it is crucial to understand the chain of reasoning, which necessarily includes reasoning from negated or speculative propositions, and claims that are backed by various amounts of potentially unreliable evidence. While the development of new interpretability methods might be developed to allow advanced users to trace models' reasoning by inspecting intermediate representations, it is perhaps more useful to end users to see a clear breakdown of how the model reaches a decision: which atomic knowledge claims make up the chain of reasoning and what precisely are the sources of evidence for (or against) each claim?

All this means that factuality is more important than ever. It seems unlikely that the use of LLMs in knowledge-based applications will cease, so we believe that it is crucial to incorporate explicit representations of factuality into them in order to bolster trust. At this point we can only speculate how our definition and representation of factuality might be incorporated to modern LLMs, especially because many of these (e.g., GPT) are transformer decoders that leave little architectural room for latent factuality representations. Still, not all of the state-of-the-art LLMs are based on the transformer decoder. For example, Aya is a 13 billion

parameter chat-focused general language model based on the encoder-decoder architecture of T5, which leaves open the possibility of utilizing latent factuality representations as we did with the BART-based VAEs.

## 6.4 Final Thoughts

This thesis focused on factualizing event extraction via the dimensions of polarity and uncertainty of knowledge claims. However, polarity and uncertainty are only summaries of the evidence according to the author, and there are myriad other context dimensions that are crucial for determining claim factuality. As we mentioned in Chapter 1, the full context of a knowledge claim is precisely the entirety of the supporting evidence presented in the parent study, as well as the parent study's network of citations. Thus, even if the author expresses a claim with high certainty, it may be that other context dimensions such as data quality, statistical methods, and the reliability of cited work betray a less certain conclusion. There has been significant previous work on extracting context dimensions (cf. Chapter 2), but an important line of future research is to determine how values of these myriad context dimensions affect overall factuality.

Importantly, while we focused on biomedical literature, the methods and insights developed in this thesis are equally applicable to other domains such as news, social media, finance, and law. All of these domains already benefit from NLP for extracting and summarizing events, and the ability to determine their polarity and uncertainty would be helpful when using NLP tools to make informed decisions. For example, the relatively new NLP tasks of misinformation detection and automated fact-checking are necessarily concerned with the factuality of events and so are a natural application area for the methods and models we developed here. Still, the proposed conceptual shift from facts to beliefs is ultimately useless if it is not adopted by researchers and practitioners. Acceptance may be difficult to foster given the relative ease of modeling only the core event representation, but we believe that continuing to champion the importance of factuality will ultimately help build a more reliable and trustworthy digital future.

# References


[1] *Is There a Reproducibility Crisis in Science?* 2016. [Online]. Available: https://www.scientificamerican.com/video/is-there-a-reproducibility-crisis-in-science/ (visited on 10/13/2019) (cited on p. 19).

[2] J. P. A. Ioannidis, "Why most published research findings are false," en, *PLOS Medicine*, vol. 2, no. 8, e124, Aug. 2005, ISSN: 1549-1676. DOI: 10.1371/journal.pmed.0020124 (cited on p. 19).

[3] P. O. Larsen and M. von Ins, "The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index.," *Scientometrics*, vol. 84, no. 3, pp. 575–603, Sep. 2010, ISSN: 0138-9130. DOI: 10.1007/s11192-010-0202-z. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20700371%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2909426 (cited on p. 19).

[4] P. Thompson, R. Nawaz, J. McNaught, and S. Ananiadou, "Enriching a biomedical event corpus with meta-knowledge annotation," *BMC bioinformatics*, vol. 12, no. 1, pp. 1–18, 2011 (cited on pp. 19, 45).

[5] P. Thompson, R. Nawaz, J. McNaught, and S. Ananiadou, "Enriching news events with meta-knowledge information," *Lang. Resour. Evaluation*, vol. 51, no. 2, pp. 409–438, 2017. DOI: 10.1007/s10579-016-9344-9. [Online]. Available: https://doi.org/10.1007/s10579-016-9344-9 (cited on p. 19).

[6] N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky, "Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, S. Manandhar and D. Yuret, Eds., Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 1–9. [Online]. Available: https://aclanthology.org/S13-2001 (cited on p. 19).

[7] G. Szarvas, V. Vincze, R. Farkas, and J. Csirik, "The bioscope corpus: Annotation for negation, uncertainty and their scope in biomedical texts," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*,

ser. BioNLP '08, USA: Association for Computational Linguistics, Jun. 2008, pp. 38–45, ISBN: 978-1-932432-11-4 (cited on pp. 20, 43).

[8] M. Light, X. Y. Qiu, and P. Srinivasan, "The language of bioscience: Facts, speculations, and statements in between," in *HLT-NAACL 2004 Workshop: Linking Biological Literature, Ontologies and Databases*, Association for Computational Linguistics, May 2004, pp. 17–24. [Online]. Available: https://www.aclweb.org/anthology/W04-3103 (cited on p. 20).

[9] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *Journal of Biomedical Informatics*, vol. 34, no. 5, pp. 301–310, Oct. 2001, ISSN: 1532-0464. DOI: 10.1006/jbin.2001.1029 (cited on pp. 20, 40).

[10] NIST, "The ace 2005 (ace05) evaluation plan," 2005 (cited on pp. 20, 27, 31).

[11] R. Saurı and T. Badia, "Catalan timebank 1.0 corpus documentation," Technical report, Tech. Rep., 2012. [Online]. Available: https://catalog.ldc.upenn.edu/docs/LDC2012T10/README_catalanTimeBank.pdf (cited on p. 20).

[12] S. Zheng, W. Cao, W. Xu, and J. Bian, "Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 337–346. DOI: 10.18653/v1/D19-1032. [Online]. Available: https://aclanthology.org/D19-1032 (cited on p. 20).

[13] Z. Zhou, L. Ma, and H. Liu, "Trade the event: Corporate events detection for news-based event-driven trading," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 2114–2124. DOI: 10.18653/v1/2021.findings-acl.186. [Online]. Available: https://aclanthology.org/2021.findings-acl.186 (cited on p. 20).

[14] A. Colas, A. Sadeghian, Y. Wang, and D. Z. Wang, *Eventnarrative: A large-scale event-centric dataset for knowledge graph-to-text generation*, 2022. arXiv: 2111.00276 [cs.CL] (cited on p. 20).

[15] A. Bittar, P. Amsili, P. Denis, and L. Danlos, "French TimeBank: An ISO-TimeML annotated reference corpus," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds., Portland, Oregon, USA: Association for

Computational Linguistics, Jun. 2011, pp. 130–134. [Online]. Available: https://aclanthology.org/P11-2023 (cited on p. 20).

[16] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "Genia corpus—a semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, no. suppl_1, pp. i180–i182, 2003 (cited on pp. 20, 121).

[17] A. Halterman, K. Keith, S. Sarwar, and B. O'Connor, "Corpus-level evaluation for event QA: The IndiaPoliceEvents corpus covering the 2002 Gujarat violence," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 4240–4253. DOI: 10.18653/v1/2021.findings-acl.371. [Online]. Available: https://aclanthology.org/2021.findings-acl.371 (cited on p. 20).

[18] M. Li, A. Zareian, Q. Zeng, *et al.*, "Cross-media structured common space for multimedia event extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 2557–2568. DOI: 10.18653/v1/2020.acl-main.230. [Online]. Available: https://aclanthology.org/2020.acl-main.230 (cited on p. 20).

[19] Z. Sun, J. Li, G. Pergola, *et al.*, *Phee: A dataset for pharmacovigilance event extraction from text*, 2022. arXiv: 2210.12560 [cs.CL] (cited on p. 20).

[20] S. Jain, M. van Zuylen, H. Hajishirzi, and I. Beltagy, "SciREX: A challenge dataset for document-level information extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 7506–7516. DOI: 10.18653/v1/2020.acl-main.670. [Online]. Available: https://aclanthology.org/2020.acl-main.670 (cited on p. 20).

[21] J. PUSTEJOVSKY, "The timebank corpus," in *Proceedings of Corpus Linguistics 2003*, 2003, pp. 647–656 (cited on p. 20).

[22] H. Deng, Y. Zhang, Y. Zhang, *et al.*, *Title2event: Benchmarking open event extraction with a large-scale chinese title dataset*, 2022. arXiv: 2211.00869 [cs.CL] (cited on p. 20).

[23] S. Li, H. Ji, and J. Han, "Document-level event argument extraction by conditional generation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Online: Association for

Computational Linguistics, Jun. 2021, pp. 894–908. DOI: 10.18653/v1/2021.naacl-main.69. [Online]. Available: https://aclanthology.org/2021.naacl-main.69 (cited on p. 20).

[24] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of biomedical informatics*, vol. 45, no. 5, pp. 885–892, 2012 (cited on p. 20).

[25] L. Luo, P.-T. Lai, C.-H. Wei, C. N. Arighi, and Z. Lu, "Biored: A rich biomedical relation extraction dataset," *Briefings in Bioinformatics*, vol. 23, no. 5, bbac282, 2022 (cited on p. 20).

[26] O. Taboureau, S. K. Nielsen, K. Audouze, *et al.*, "Chemprot: A disease chemical biology database," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D367–D372, 2010 (cited on p. 20).

[27] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, S. Manandhar and D. Yuret, Eds., Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 341–350. [Online]. Available: https://aclanthology.org/S13-2056 (cited on p. 20).

[28] Y. Yao, D. Ye, P. Li, *et al.*, "DocRED: A large-scale document-level relation extraction dataset," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 764–777. DOI: 10.18653/v1/P19-1074. [Online]. Available: https://aclanthology.org/P19-1074 (cited on p. 20).

[29] X. Han, H. Zhu, P. Yu, *et al.*, "FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4803–4809. DOI: 10.18653/v1/D18-1514. [Online]. Available: https://aclanthology.org/D18-1514 (cited on p. 20).

[30] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/va challenge on concepts, assertions, and relations in clinical text," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 552–556, 2011 (cited on p. 20).

[31] S. Jain, A. Agrawal, A. Saporta, *et al.*, *Radgraph: Extracting clinical entities and relations from radiology reports*, 2021. arXiv: 2106.14463 [cs.CL] (cited on p. 20).

[32] Y. Luan, L. He, M. Ostendorf, and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3219–3232. DOI: 10.18653/v1/D18-1360. [Online]. Available: https://aclanthology.org/D18-1360 (cited on pp. 20, 24).

[33] I. Hendrickx, S. N. Kim, Z. Kozareva, *et al.*, "SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, K. Erk and C. Strapparava, Eds., Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 33–38. [Online]. Available: https://aclanthology.org/S10-1006 (cited on p. 20).

[34] H. Elsahar, P. Vougiouklis, A. Remaci, *et al.*, "T-REx: A large scale alignment of natural language with knowledge base triples," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. Calzolari, K. Choukri, C. Cieri, *et al.*, Eds., Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1544 (cited on p. 20).

[35] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 35–45. DOI: 10.18653/v1/D17-1004. [Online]. Available: https://aclanthology.org/D17-1004 (cited on p. 20).

[36] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "Creating training corpora for NLG micro-planners," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds., Vancouver, Canada: Association for Computational Linguistics,

Jul. 2017, pp. 179–188. DOI: 10.18653/v1/P17-1017. [Online]. Available: https://aclanthology.org/P17-1017 (cited on p. 20).

[37] V. Basile, *The perspectivist data manifesto*. [Online]. Available: https://pdai.info/ (cited on pp. 21, 61, 94).

[38] A. Kratzer, "Facts: Particulars or information units?" *Linguistics and Philosophy*, vol. 25, no. 5/6, pp. 655–670, 2002, ISSN: 0165-0157 (cited on pp. 21, 22).

[39] H. Kilicoglu, G. Rosemblat, and T. C. Rindflesch, "Assigning factuality values to semantic relations extracted from biomedical research literature," en, *PLOS ONE*, vol. 12, no. 7, e0179926, Jul. 2017, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0179926 (cited on pp. 23, 42, 43, 49, 50, 54–56, 60).

[40] A. Jøsang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty* (Artificial Intelligence: Foundations, Theory, and Algorithms), English. Switzerland: Springer International Publishing, 2016, ISBN: 978-3-319-42335-7 (cited on pp. 24, 57).

[41] M.-C. de Marneffe, C. D. Manning, and C. Potts, "Did it happen? the pragmatic complexity of veridicality assessment," *Computational Linguistics*, vol. 38, no. 2, pp. 301–333, Jun. 2012, ISSN: 0891-2017. DOI: 10.1162/COLI_a_00097 (cited on p. 24).

[42] A. Cohan, G. Feigenblat, D. Freitag, *et al.*, "Overview of the third workshop on scholarly document processing," in *Proceedings of the Third Workshop on Scholarly Document Processing*, A. Cohan, G. Feigenblat, D. Freitag, *et al.*, Eds., Gyeongju, Republic of Korea: Association for Computational Linguistics, Oct. 2022, pp. 1–6. [Online]. Available: https://aclanthology.org/2022.sdp-1.1 (cited on p. 24).

[43] M. K. Chandrasekaran, M. Yasunaga, D. Radev, D. Freitag, and M.-Y. Kan, "Overview and results: Cl-scisumm shared task 2019," no. arXiv:1907.09854, Jul. 2019, arXiv:1907.09854 [cs]. DOI: 10.48550/arXiv.1907.09854. [Online]. Available: http://arxiv.org/abs/1907.09854 (cited on p. 24).

[44] O. Bodenreider, "The unified medical language system (umls): Integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. $\text{suppl}_1$, pp. D267–D270, Jan. 2004, ISSN: 0305-1048. DOI: 10.1093/nar/gkh061 (cited on pp. 24, 30, 49).

[45] e. a. Donnelly Kevin, "Snomed-ct: The advanced terminology and coding system for ehealth," *Studies in health technology and informatics*, vol. 121, p. 279, 2006 (cited on p. 24).

[46] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez gene: Gene-centered information at ncbi," *Nucleic acids research*, vol. 39, no. suppl_1, pp. D52–D57, 2010 (cited on p. 24).

[47] R. F. Rizvi, J. Vasilakes, T. J. Adam, *et al.*, "Idisk: The integrated dietary supplements knowledge base," *Journal of the American Medical Informatics Association*, vol. 27, no. 4, pp. 539–548, 2020 (cited on p. 24).

[48] J. S. Ash, D. F. Sittig, E. M. Campbell, K. P. Guappone, and R. H. Dykstra, "Some unintended consequences of clinical decision support systems," en, *AMIA Annual Symposium Proceedings*, vol. 2007, p. 26, 2007 (cited on p. 24).

[49] T. Payne, *Exploring language structure: A student's guide*. Cambridge University Press, 2006 (cited on p. 27).

[50] C. Friedman, P. Kra, and A. Rzhetsky, "Two biomedical sublanguages: A description based on the theories of zellig harris," *Journal of Biomedical Informatics*, Sublanguage - Zellig Harris Memorial, vol. 35, no. 4, pp. 222–235, Aug. 2002, ISSN: 1532-0464. DOI: 10.1016/S1532-0464(03)00012-1 (cited on pp. 28, 83).

[51] D. S. Carrell, R. E. Schoen, D. A. Leffler, *et al.*, "Challenges in adapting existing clinical natural language processing systems to multiple, diverse health care settings," *Journal of the American Medical Informatics Association : JAMIA*, vol. 24, no. 5, pp. 986–991, Apr. 2017, ISSN: 1067-5027. DOI: 10.1093/jamia/ocx039 (cited on p. 28).

[52] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," en, 2018. [Online]. Available: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (cited on pp. 29, 37).

[53] R. Grishman, S. Huttunen, and R. Yangarber, "Information extraction for enhanced access to disease outbreak reports," *Journal of Biomedical Informatics*, Sublanguage - Zellig Harris Memorial, vol. 35, no. 4, pp. 236–246, Aug. 2002, ISSN: 1532-0464. DOI: 10.1016/S1532-0464(03)00013-3 (cited on p. 30).

[54] T. C. Rindflesch and M. Fiszman, "The interaction of domain knowledge and linguistic structure in natural language processing: Interpreting hypernymic propositions in biomedical text," *Journal of Biomedical Informatics*, Unified Medical Language System, vol. 36, no. 6, pp. 462–477, Dec. 2003, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2003.11.003 (cited on p. 30).

[55] T. C. Rindflesch, M. Fiszman, and B. Libbus, "Semantic interpretation for the biomedical research literature," en, in *Medical Informatics: Knowledge Management and Data Mining in Biomedicine* (Integrated Series in Information Systems), H. Chen,

S. S. Fuller, C. Friedman, and W. Hersh, Eds., Integrated Series in Information Systems. Boston, MA: Springer US, 2005, pp. 399–422, ISBN: 978-0-387-25739-6. DOI: 10.1007/0-387-25739-X_14. [Online]. Available: https://doi.org/10.1007/0-387-25739-X_14 (cited on p. 30).

[56] H. Kilicoglu, G. Rosemblat, M. Fiszman, and D. Shin, "Broad-coverage biomedical relation extraction with semrep," *BMC Bioinformatics*, vol. 21, no. 1, p. 188, May 2020, ISSN: 1471-2105. DOI: 10.1186/s12859-020-3517-7 (cited on pp. 30, 49, 52).

[57] K. B. Cohen, K. Verspoor, H. Johnson, *et al.*, "High-precision biological event extraction with a concept recognizer," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 50–58. [Online]. Available: https://aclanthology.org/W09-1407 (cited on pp. 30, 41).

[58] Q.-C. Bui and P. M. Sloot, "A robust approach to extract biomedical events from literature," *Bioinformatics*, vol. 28, no. 20, pp. 2654–2661, Oct. 2012, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts487 (cited on p. 30).

[59] Q.-C. Bui, D. Campos, E. van Mulligen, and J. Kors, "A fast rule-based approach for biomedical event extraction," in *Proceedings of the BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 104–108. [Online]. Available: https://aclanthology.org/W13-2014 (cited on p. 30).

[60] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of bionlp'09 shared task on event extraction," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 1–9. [Online]. Available: https://aclanthology.org/W09-1401 (cited on pp. 30, 43).

[61] J. Björne, J. Heimonen, F. Ginter, A. Airola, T. Pahikkala, and T. Salakoski, "Extracting complex biological events with rich graph-based feature sets," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 10–18. [Online]. Available: https://aclanthology.org/W09-1402 (cited on p. 30).

[62] R. Roller and M. Stevenson, "Identification of genia events using multiple classifiers," in *Proceedings of the BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 125–129. [Online]. Available: https://aclanthology.org/W13-2018 (cited on p. 30).

[63] M. Miwa, R. Sætre, J.-D. Kim, and J. Tsujii, "Event extraction with complex event classification using rich features," *Journal of Bioinformatics and Computational Biology*, vol. 08, no. 01, pp. 131–146, Feb. 2010, ISSN: 0219-7200. DOI: 10.1142/S0219720010004586 (cited on pp. 30, 37).

[64] M. Miwa, P. Thompson, and S. Ananiadou, "Boosting automatic event extraction from the literature using domain adaptation and coreference resolution," *Bioinformatics*, vol. 28, no. 13, pp. 1759–1765, Jul. 2012, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts237 (cited on pp. 30, 37).

[65] S. Riedel, D. McClosky, M. Surdeanu, A. McCallum, and C. D. Manning, "Model combination for event extraction in bionlp 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 51–55. [Online]. Available: https://aclanthology.org/W11-1808 (cited on p. 31).

[66] S. Riedel and A. McCallum, "Fast and robust joint models for biomedical event extraction," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 1–12. [Online]. Available: https://aclanthology.org/D11-1001 (cited on p. 31).

[67] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 73–82. [Online]. Available: https://aclanthology.org/P13-1008 (cited on p. 31).

[68] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, Association for Computational Linguistics, Jul. 2002, pp. 1–8. DOI: 10.3115/1118693.1118694. [Online]. Available: https://aclanthology.org/W02-1001 (cited on p. 31).

[69] Q. Li and H. Ji, "Incremental joint extraction of entity mentions and relations," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 402–412. DOI: 10.3115/v1/P14-1038. [Online]. Available: https://aclanthology.org/P14-1038 (cited on pp. 31, 32).

[70] H. Lin and S. Jegelka, "Resnet with one-neuron hidden layers is a universal approximator," in *Advances in Neural Information Processing Systems*, vol. 31, Curran

Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/03bfc1d4783966c69cc6aef8247e0103-Abstract.html (cited on p. 31).

[71] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," no. arXiv:1301.3781, Sep. 2013, arXiv:1301.3781 [cs]. DOI: 10.48550/arXiv.1301.3781. [Online]. Available: http://arxiv.org/abs/1301.3781 (cited on pp. 31, 32).

[72] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. [Online]. Available: https://aclanthology.org/D14-1162 (cited on p. 31).

[73] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, L. Vanderwende, H. Daumé III, and K. Kirchhoff, Eds., Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 746–751. [Online]. Available: https://aclanthology.org/N13-1090 (cited on p. 32).

[74] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Toutanova and H. Wu, Eds., Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 238–247. DOI: 10.3115/v1/P14-1023. [Online]. Available: https://aclanthology.org/P14-1023 (cited on p. 32).

[75] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, M. Collins and L. Lee, Eds., pp. 211–225, 2015. DOI: 10.1162/tacl_a_00134 (cited on p. 32).

[76] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, no. null, pp. 2493–2537, Nov. 2011, ISSN: 1532-4435 (cited on p. 32).

[77] C. Quan, L. Hua, X. Sun, and W. Bai, "Multichannel convolutional neural network for biological relation extraction," *BioMed Research International*, vol. 2016, p. 1 850 404, 2016, ISSN: 2314-6133. DOI: 10.1155/2016/1850404 (cited on p. 32).

[78] J. Björne and T. Salakoski, "Biomedical event extraction using convolutional neural networks and dependency parsing," in *Proceedings of the BioNLP 2018 workshop*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 98–108. DOI: 10.18653/v1/W18-2311. [Online]. Available: https://aclanthology.org/W18-2311 (cited on p. 32).

[79] A. Wang, J. Wang, H. Lin, J. Zhang, Z. Yang, and K. Xu, "A multiple distributed representation method based on neural network for biomedical event extraction," en, *BMC Medical Informatics and Decision Making*, vol. 17, no. 3, p. 171, Dec. 2017, ISSN: 1472-6947. DOI: 10.1186/s12911-017-0563-9 (cited on p. 32).

[80] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 300–309. DOI: 10.18653/v1/N16-1034. [Online]. Available: https://aclanthology.org/N16-1034 (cited on p. 32).

[81] M. Miwa and M. Bansal, "End-to-end relation extraction using lstms on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1105–1116. DOI: 10.18653/v1/P16-1105. [Online]. Available: https://aclanthology.org/P16-1105 (cited on p. 32).

[82] P. Zhou, W. Shi, J. Tian, *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 207–212. DOI: 10.18653/v1/P16-2034. [Online]. Available: https://aclanthology.org/P16-2034 (cited on pp. 32–34).

[83] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," no. arXiv:1409.0473, May 2016, arXiv:1409.0473 [cs, stat]. DOI: 10.48550/arXiv.1409.0473. [Online]. Available: http://arxiv.org/abs/1409.0473 (cited on p. 32).

[84] S. Rao, D. Marcu, K. Knight, and H. Daumé III, "Biomedical event extraction using abstract meaning representation," in *BioNLP 2017*, Vancouver, Canada, Association for Computational Linguistics, Aug. 2017, pp. 126–135. DOI: 10.18653/v1/W17-2315. [Online]. Available: https://aclanthology.org/W17-2315 (cited on p. 33).

[85] L. Banarescu, C. Bonial, S. Cai, *et al.*, "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 178–186. [Online]. Available: https://aclanthology.org/W13-2322 (cited on p. 33).

[86] T.-J. Fu, P.-H. Li, and W.-Y. Ma, "Graphrel: Modeling text as relational graphs for joint entity and relation extraction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1409–1418. DOI: 10.18653/v1/P19-1136. [Online]. Available: https://www.aclweb.org/anthology/P19-1136 (cited on p. 33).

[87] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017 (cited on p. 33).

[88] F. Christopoulou, M. Miwa, and S. Ananiadou, "A walk-based model on entity graphs for relation extraction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 81–88. DOI: 10.18653/v1/P18-2014. [Online]. Available: https://aclanthology.org/P18-2014 (cited on p. 33).

[89] F. Christopoulou, M. Miwa, and S. Ananiadou, "Connecting the dots: Document-level neural relation extraction with edge-oriented graphs," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4925–4936. DOI: 10.18653/v1/D19-1498. [Online]. Available: https://aclanthology.org/D19-1498 (cited on pp. 33, 34).

[90] F. Christopoulou, T. T. Tran, S. K. Sahu, M. Miwa, and S. Ananiadou, "Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods," *Journal of the American Medical Informatics Association*,

vol. 27, no. 1, pp. 39–46, Jan. 2019. DOI: https://doi.org/10.1093/jamia/ocz101 (cited on p. 34).

[91] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 6000–6010, ISBN: 978-1-5108-6096-4 (cited on pp. 34–36, 83).

[92] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, and X. Carreras, Eds., Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 551–561. DOI: 10.18653/v1/D16-1053. [Online]. Available: https://aclanthology.org/D16-1053 (cited on p. 35).

[93] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423 (cited on p. 36).

[94] I. Beltagy, K. Lo, and A. Cohan, "Scibert: Pretrained language model for scientific text," in *EMNLP*, 2019. eprint: arXiv:1903.10676 (cited on p. 36).

[95] J. Lee, W. Yoon, S. Kim, *et al.*, "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020 (cited on p. 36).

[96] Y. Gu, R. Tinn, H. Cheng, *et al.*, *Domain-specific language model pretraining for biomedical natural language processing*, 2020. eprint: arXiv:2007.15779 (cited on p. 36).

[97] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," in *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, 2019, pp. 58–65 (cited on p. 36).

[98] E. Alsentzer, J. Murphy, W. Boag, *et al.*, "Publicly available clinical BERT embeddings," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 72–78. DOI: 10.18653/v1/W19-1909 (cited on p. 36).

[99] Y. Goldberg, "Assessing bert's syntactic abilities," no. arXiv:1901.05287, Jan. 2019, arXiv:1901.05287 [cs]. DOI: 10.48550/arXiv.1901.05287. [Online]. Available: http://arxiv.org/abs/1901.05287 (cited on p. 36).

[100] G. Jawahar, B. Sagot, and D. Seddah, "What does bert learn about the structure of language?" In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3651–3657. DOI: 10.18653/v1/P19-1356. [Online]. Available: https://aclanthology.org/P19-1356 (cited on p. 36).

[101] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2895–2905. DOI: 10.18653/v1/P19-1279. [Online]. Available: https://www.aclweb.org/anthology/P19-1279 (cited on pp. 36, 37, 99).

[102] Z. Zhong and D. Chen, "A frustratingly easy approach for entity and relation extraction," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 50–61. DOI: 10.18653/v1/2021.naacl-main.5. [Online]. Available: https://aclanthology.org/2021.naacl-main.5 (cited on pp. 36, 99).

[103] D. Ye, Y. Lin, P. Li, and M. Sun, "Packed levitated marker for entity and relation extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4904–4917. DOI: 10.18653/v1/2022.acl-long.337 (cited on pp. 37, 99, 101).

[104] Q. Wei, Z. Ji, Y. Si, *et al.*, "Relation extraction from clinical narratives using pre-trained language models," *AMIA Annual Symposium Proceedings*, vol. 2019, pp. 1236–1245, Mar. 2020, ISSN: 1942-597X (cited on p. 37).

[105] A. Ramponi, R. van der Goot, R. Lombardo, and B. Plank, "Biomedical event extraction as sequence labeling," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online: Association for Computational Linguistics, Nov. 2020, pp. 5357–5367. DOI: 10.18653/v1/2020.emnlp-main.431. [Online]. Available: https://aclanthology.org/2020.emnlp-main.431 (cited on p. 37).

[106] H.-L. Trieu, T. T. Tran, K. N. A. Duong, A. Nguyen, M. Miwa, and S. Ananiadou, "Deepeventmine: End-to-end neural nested event extraction from biomedical texts," *Bioinformatics*, vol. 36, no. 19, pp. 4910–4917, Dec. 2020, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa540 (cited on pp. 37, 107).

[107] P. J. Liu*, M. Saleh*, E. Pot, *et al.*, "Generating wikipedia by summarizing long sequences," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Hyg0vbWC- (cited on p. 37).

[108] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019. [Online]. Available: http://www.persagen.com/files/misc/radford2019language.pdf (cited on pp. 37, 75, 79).

[109] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, 140:5485–140:5551, Jan. 2020, ISSN: 1532-4435 (cited on p. 37).

[110] P. Lewis, M. Ott, J. Du, and V. Stoyanov, "Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art," in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, Online: Association for Computational Linguistics, Nov. 2020, pp. 146–157. DOI: 10.18653/v1/2020.clinicalnlp-1.17 (cited on pp. 37, 83).

[111] P.-L. Huguet Cabot and R. Navigli, "Rebel: Relation extraction by end-to-end language generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2370–2381. DOI: 10.18653/v1/2021.findings-emnlp.204. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.204 (cited on p. 38).

[112] R. Luo, L. Sun, Y. Xia, *et al.*, "Biogpt: Generative pre-trained transformer for biomedical text generation and mining," *Briefings in Bioinformatics*, vol. 23, no. 6, bbac409, Nov. 2022, ISSN: 1477-4054. DOI: 10.1093/bib/bbac409 (cited on p. 38).

[113] Y. Peng, X. Wang, L. Lu, M. Bagheri, R. Summers, and Z. Lu, "Negbio: A high-performance tool for negation and uncertainty detection in radiology reports," *AMIA Summits on Translational Science Proceedings*, vol. 2018, pp. 188–196, May 2018, ISSN: 2153-4063 (cited on p. 40).

[114] S. Mehrabi, A. Krishnan, S. Sohn, *et al.*, "Deepen: A negation detection system for clinical text incorporating dependency relation into negex," *Journal of biomedical informatics*, vol. 54, pp. 213–219, 2015 (cited on p. 40).

[115] R. Morante and W. Daelemans, "A metalearning approach to processing the scope of negation," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, S. Stevenson and X. Carreras, Eds., Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 21–29. [Online]. Available: https://aclanthology.org/W09-1105 (cited on pp. 40, 98).

[116] R. Morante and W. Daelemans, "Learning the scope of hedge cues in biomedical texts," in *Proceedings of the BioNLP 2009 Workshop*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 28–36. [Online]. Available: https://aclanthology.org/W09-1304 (cited on pp. 40, 98).

[117] S. Agarwal and H. Yu, "Detecting hedge cues and their scope in biomedical text with conditional random fields," *Journal of Biomedical Informatics*, vol. 43, no. 6, pp. 953–961, Dec. 2010, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2010.08.003 (cited on p. 40).

[118] E. Velldal, L. Øvrelid, J. Read, and S. Oepen, "Speculation and negation: Rules, rankers, and the role of syntax," *Computational Linguistics*, vol. 38, no. 2, pp. 369–410, Jun. 2012. DOI: 10.1162/COLI_a_00126 (cited on p. 40).

[119] B. Zou, G. Zhou, and Q. Zhu, "Tree kernel-based negation and speculation scope detection with structured syntactic parse features," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 968–976. [Online]. Available: https://aclanthology.org/D13-1099 (cited on p. 40).

[120] F. Fancellu, A. Lopez, and B. Webber, "Neural networks for negation scope detection," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Aug. 2016, pp. 495–504. DOI: 10.18653/v1/P16-1047. [Online]. Available: https://www.aclweb.org/anthology/P16-1047 (cited on pp. 40, 98).

[121] R. Morante and E. Blanco, "*sem 2012 shared task: Resolving the scope and focus of negation," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, Montréal, Canada: Association for Computational Linguistics, Jul. 2012, pp. 265–274. [Online]. Available: https://aclanthology.org/S12-1035 (cited on pp. 40, 43).

[122] Z. Qian, P. Li, Q. Zhu, G. Zhou, Z. Luo, and W. Luo, "Speculation and negation scope detection via convolutional neural networks," in *Proceedings of the 2016*

*Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 815–825. DOI: 10.18653/v1/D16-1078. [Online]. Available: https://aclanthology.org/D16-1078 (cited on p. 40).

[123] E. Sergeeva, H. Zhu, A. Tahmasebi, and P. Szolovits, "Neural token representations and negation and speculation scope detection in biomedical and general domain text," in *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 178–187. DOI: 10.18653/v1/D19-6221. [Online]. Available: https://aclanthology.org/D19-6221 (cited on p. 41).

[124] C. Lin, S. Bethard, D. Dligach, F. Sadeque, G. Savova, and T. A. Miller, "Does BERT need domain adaptation for clinical negation detection?" *Journal of the American Medical Informatics Association*, vol. 27, no. 4, pp. 584–591, Apr. 2020. DOI: 10.1093/jamia/ocaa001 (cited on p. 41).

[125] A. Khandelwal and S. Sawant, "NegBERT: A transfer learning approach for negation detection and scope resolution," in *Proceedings of the 12th Language Resources and Evaluation Conference*, European Language Resources Association, May 2020, pp. 5739–5748, ISBN: 979-10-95546-34-4. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.704 (cited on pp. 41, 98).

[126] A. Khandelwal and B. K. Britto, "Multitask learning of negation and speculation using transformers," in *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, Online: Association for Computational Linguistics, Nov. 2020, pp. 79–87. DOI: 10.18653/v1/2020.louhi-1.9. [Online]. Available: https://aclanthology.org/2020.louhi-1.9 (cited on p. 41).

[127] P. Tiwary, A. Madhubalan, and A. Gautam, "No means 'no': A non-improper modeling approach, with embedded speculative context," *Bioinformatics*, vol. 38, no. 20, pp. 4790–4796, Oct. 2022, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac593 (cited on p. 41).

[128] K. r. Kanakarajan, B. Kundumani, and M. Sankarasubbu, "Bioelectra:pretrained biomedical text encoder using discriminators," in *Proceedings of the 20th Workshop on Biomedical Language Processing*, Online: Association for Computational Linguistics, Jun. 2021, pp. 143–154. DOI: 10.18653/v1/2021.bionlp-1.16. [Online]. Available: https://aclanthology.org/2021.bionlp-1.16 (cited on p. 41).

[129] F. Sarafraz and G. Nenadic, "Using svms with the command relation features to identify negated events in biomedical literature," in *Proceedings of the Workshop on*

*Negation and Speculation in Natural Language Processing*, Uppsala, Sweden: University of Antwerp, Jul. 2010, pp. 78–85. [Online]. Available: https://aclanthology.org/W10-3113 (cited on p. 41).

[130] A. MacKinlay, D. Martinez, and T. Baldwin, "Biomedical event annotation with crfs and precision grammars," in *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 77–85. [Online]. Available: https://aclanthology.org/W09-1410 (cited on p. 41).

[131] M. Miwa, P. Thompson, J. McNaught, D. B. Kell, and S. Ananiadou, "Extracting semantically enriched events from biomedical literature," *BMC Bioinformatics*, vol. 13, no. 1, p. 108, May 2012, ISSN: 1471-2105. DOI: 10.1186/1471-2105-13-108 (cited on pp. 41, 42).

[132] H. Kilicoglu, G. Rosemblat, M. Cairelli, and T. Rindflesch, "A compositional interpretation of biomedical event factuality," in *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 22–31. DOI: 10.3115/v1/W15-1303. [Online]. Available: https://aclanthology.org/W15-1303 (cited on p. 41).

[133] C. Zerva, "Automatic identification of textual uncertainty," Ph.D. dissertation, University of Manchester, 2019. [Online]. Available: https://www.escholar.manchester.ac.uk/uk-ac-man-scw:318549 (cited on pp. 42, 61, 73, 94, 119).

[134] D. Mahajan, J. J. Liang, C.-H. Tsou, and Ö. Uzuner, "Overview of the 2022 n2c2 shared task on contextualized medication event extraction in clinical notes," *Journal of Biomedical Informatics*, vol. 144, p. 104 432, Aug. 2023, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104432 (cited on pp. 42, 96).

[135] T. Tsujimura, K. Yamada, R. Ida, M. Miwa, and Y. Sasaki, "Contextualized medication event extraction with striding ner and multi-turn qa," *Journal of Biomedical Informatics*, vol. 144, p. 104 416, Aug. 2023, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104416 (cited on p. 42).

[136] P. Thompson, R. Nawaz, J. McNaught, and S. Ananiadou, "Enriching a biomedical event corpus with meta-knowledge annotation," *BMC Bioinformatics*, vol. 12, no. 1, p. 393, Oct. 2011, ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-393 (cited on pp. 43, 50).

[137] D. Mahajan, J. J. Liang, and C.-H. Tsou, "Toward understanding clinical context of medication change events in clinical narratives," *AMIA Annual Symposium Proceedings*, vol. 2021, pp. 833–842, Feb. 2022, ISSN: 1942-597X (cited on pp. 43, 44, 99).

[138] N. Konstantinova, S. C. de Sousa, N. P. Cruz, M. J. Maña, M. Taboada, and R. Mitkov, "A review corpus annotated for negation, speculation and their scope," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 3190–3195. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/533_Paper.pdf (cited on pp. 43, 101).

[139] P. Thompson, R. Nawaz, J. McNaught, and S. Ananiadou, "Enriching news events with meta-knowledge information," en, *Language Resources and Evaluation*, vol. 51, no. 2, pp. 409–438, Jun. 2017, ISSN: 1574-0218. DOI: 10.1007/s10579-016-9344-9 (cited on p. 43).

[140] R. Saurí and J. Pustejovsky, "Factbank: A corpus annotated with event factuality," *Lang. Resour. Evaluation*, vol. 43, no. 3, pp. 227–268, 2009. DOI: 10.1007/s10579-009-9089-9. [Online]. Available: https://doi.org/10.1007/s10579-009-9089-9 (cited on p. 43).

[141] C. Dalloux, V. Claveau, and N. Grabar, "Speculation and negation detection in french biomedical corpora," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, Varna, Bulgaria: INCOMA Ltd., Sep. 2019, pp. 223–232. DOI: 10.26615/978-954-452-056-4_026. [Online]. Available: https://www.aclweb.org/anthology/R19-1026 (cited on p. 43).

[142] T. Kang, S. Zhang, N. Xu, D. Wen, X. Zhang, and J. Lei, "Detecting negation and scope in chinese clinical notes using character and word embedding," *Computer Methods and Programs in Biomedicine*, vol. 140, pp. 53–59, Mar. 2017, ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2016.11.009 (cited on p. 43).

[143] S. Lima Lopez, N. Perez, M. Cuadros, and G. Rigau, "Nubes: A corpus of negation and uncertainty in spanish clinical texts," English, in *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, May 2020, pp. 5772–5781, ISBN: 979-10-95546-34-4. [Online]. Available: https://aclanthology.org/2020.lrec-1.708 (cited on p. 43).

[144] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009 (cited on p. 46).

[145] X. J. Zhu, "Semi-supervised learning literature survey," 2005 (cited on p. 46).

[146] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, ISSN: 1558-2191. DOI: 10.1109/TKDE.2009.191 (cited on p. 46).

[147] P. Ren, Y. Xiao, X. Chang, *et al.*, "A survey of deep active learning," *ACM Computing Surveys*, vol. 54, no. 9, 180:1–180:40, Oct. 2021, ISSN: 0360-0300. DOI: 10.1145/3472291 (cited on p. 46).

[148] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," en, *Machine Learning*, vol. 109, no. 2, pp. 373–440, Feb. 2020, ISSN: 1573-0565. DOI: 10.1007/s10994-019-05855-6 (cited on p. 46).

[149] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, Oct. 2020, ISSN: 2691-4581. DOI: 10.1109/TAI.2021.3054609 (cited on p. 46).

[150] K. Wang, Y. Qiu, Y. Hong, *et al.*, "A case study on active learning for event extraction," en, in *Social Media Processing* (Communications in Computer and Information Science), Y. Li, G. Xiang, H. Lin, and M. Wang, Eds., Communications in Computer and Information Science. Singapore: Springer Singapore, 2016, vol. 669, pp. 134–142, ISBN: 978-981-10-2992-9. DOI: 10.1007/978-981-10-2993-6_11. [Online]. Available: http://link.springer.com/10.1007/978-981-10-2993-6_11 (cited on p. 46).

[151] H. Sun and R. Grishman, "Employing lexicalized dependency paths for active learning of relation extraction," en, *Intelligent Automation  Soft Computing*, vol. 34, no. 3, pp. 1415–1423, 2022, ISSN: 1079-8587. DOI: 10.32604/iasc.2022.030794 (cited on p. 46).

[152] J. Vasilakes, R. Rizvi, G. B. Melton, S. Pakhomov, and R. Zhang, "Evaluating active learning methods for annotating semantic predications," *JAMIA Open*, vol. 1, no. 2, pp. 275–282, Oct. 2018, ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooy021 (cited on pp. 47, 48, 51, 52).

[153] J. Ferguson, C. Lockard, D. Weld, and H. Hajishirzi, "Semi-supervised event extraction with paraphrase clusters," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, M. Walker, H. Ji, and A. Stent, Eds., New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 359–364. DOI: 10.18653/v1/N18-2058. [Online]. Available: https://aclanthology.org/N18-2058 (cited on p. 47).

[154] J. Wang, Q. Xu, H. Lin, Z. Yang, and Y. Li, "Semi-supervised method for biomedical event extraction," *Proteome Science*, vol. 11, no. 1, S17, Nov. 2013, ISSN: 1477-5956. DOI: 10.1186/1477-5956-11-S1-S17 (cited on p. 47).

[155] D. Zhou and D. Zhong, "A semi-supervised learning framework for biomedical event extraction based on hidden topics," *Artificial Intelligence in Medicine*, vol. 64, no. 1, pp. 51–58, May 2015, ISSN: 0933-3657. DOI: 10.1016/j.artmed.2015.03.004 (cited on p. 47).

[156] S. Pyysalo, T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou, "Event extraction across multiple levels of biological organization," *Bioinformatics*, vol. 28, no. 18, pp. i575–i581, Sep. 2012, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts407 (cited on p. 47).

[157] M. Song, H. Yu, and W.-S. Han, "Combining active learning and semi-supervised learning techniques to extract protein interaction sentences," en, *BMC Bioinformatics*, vol. 12, no. 12, S4, Nov. 2011, ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-S12-S4 (cited on p. 47).

[158] Y. Lu, H. Lin, J. Xu, *et al.*, "Text2event: Controllable sequence-to-structure generation for end-to-end event extraction," en, in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, 2021, pp. 2795–2806. DOI: 10.18653/v1/2021.acl-long.217. [Online]. Available: https://aclanthology.org/2021.acl-long.217 (cited on p. 47).

[159] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," no. arXiv:1910.10683, Sep. 2023, arXiv:1910.10683 [cs, stat]. DOI: 10.48550/arXiv.1910.10683. [Online]. Available: http://arxiv.org/abs/1910.10683 (cited on pp. 47, 120).

[160] Q. Lyu, H. Zhang, E. Sulem, and D. Roth, "Zero-shot event extraction via transfer learning: Challenges and insights," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 322–332. DOI: 10.18653/v1/2021.acl-short.42. [Online]. Available: https://aclanthology.org/2021.acl-short.42 (cited on p. 47).

[161] Z. Zhang and H. Ji, "Abstract meaning representation guided graph encoding and decoding for joint information extraction," in *Proceedings of the 2021 Conference*

*of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 39–49. DOI: 10.18653/v1/2021.naacl-main.4. [Online]. Available: https://aclanthology.org/2021.naacl-main.4 (cited on p. 47).

[162] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, K.-Y. Su, J. Su, J. Wiebe, and H. Li, Eds., Suntec, Singapore: Association for Computational Linguistics, Aug. 2009, pp. 1003–1011. [Online]. Available: https://aclanthology.org/P09-1113 (cited on p. 47).

[163] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," en, in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds., Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, vol. 6323, pp. 148–163, ISBN: 978-3-642-15938-1. DOI: 10.1007/978-3-642-15939-8_10. [Online]. Available: http://link.springer.com/10.1007/978-3-642-15939-8_10 (cited on p. 48).

[164] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds., Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 541–550. [Online]. Available: https://aclanthology.org/P11-1055 (cited on p. 48).

[165] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds., Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1753–1762. DOI: 10.18653/v1/D15-1203. [Online]. Available: https://aclanthology.org/D15-1203 (cited on p. 48).

[166] Q. Dai, N. Inoue, P. Reisert, R. Takahashi, and K. Inui, "Distantly supervised biomedical knowledge acquisition via knowledge graph based attention," in *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications*, V. Nastase, B. Roth, L. Dietz, and A. McCallum, Eds., Minneapolis, Minnesota: Asso-

ciation for Computational Linguistics, Jun. 2019, pp. 1–10. DOI: 10.18653/v1/W19-2601. [Online]. Available: https://aclanthology.org/W19-2601 (cited on p. 48).

[167] F. Christopoulou, M. Miwa, and S. Ananiadou, "Distantly supervised relation extraction with sentence reconstruction and knowledge base priors," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova, A. Rumshisky, L. Zettlemoyer, *et al.*, Eds., Online: Association for Computational Linguistics, Jun. 2021, pp. 11–26. DOI: 10.18653/v1/2021.naacl-main.2. [Online]. Available: https://aclanthology.org/2021.naacl-main.2 (cited on p. 48).

[168] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré, "Data programming: Creating large training sets, quickly," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/6709e8d64a5f47269ed5cea9f625f7ab-Abstract.html (cited on p. 48).

[169] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 11, no. 3, pp. 269–282, Nov. 2017, ISSN: 2150-8097. DOI: 10.14778/3157794.3157797 (cited on p. 48).

[170] E. K. Mallory, M. de Rochemonteix, A. Ratner, *et al.*, "Extracting chemical reactions from text using snorkel," en, *BMC Bioinformatics*, vol. 21, no. 1, p. 217, May 2020, ISSN: 1471-2105. DOI: 10.1186/s12859-020-03542-1 (cited on p. 48).

[171] S. Amin, K. A. Dunfield, A. Vechkaeva, and G. Neumann, "A data-driven approach for noise reduction in distantly supervised biomedical relation extraction," in *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 187–194. DOI: 10.18653/v1/2020.bionlp-1.20. [Online]. Available: https://aclanthology.org/2020.bionlp-1.20 (cited on p. 48).

[172] D. Schutte, J. Vasilakes, A. Bompelli, *et al.*, "Discovering novel drug-supplement interactions using suppkg generated from the biomedical literature," *Journal of Biomedical Informatics*, vol. 131, p. 104 120, Jul. 2022, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2022.104120 (cited on pp. 48, 52).

[173] A. R. Aronson and F.-M. Lang, "An overview of metamap: Historical perspective and recent advances," *Journal of the American Medical Informatics Association*,

vol. 17, no. 3, pp. 229–236, May 2010, ISSN: 1067-5027. DOI: 10.1136/jamia.2009.002733 (cited on p. 49).

[174] H. Kilicoglu, G. Rosemblat, M. Fiszman, and T. C. Rindflesch, "Constructing a semantic predication gold standard from the biomedical literature," *BMC Bioinformatics*, vol. 12, no. 1, p. 486, Dec. 2011, ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-486 (cited on pp. 49, 51).

[175] H. Kilicoglu, D. Shin, M. Fiszman, G. Rosemblat, and T. C. Rindflesch, "Semmeddb: A pubmed-scale repository of biomedical semantic predications," *Bioinformatics*, vol. 28, no. 23, pp. 3158–3160, Dec. 2012, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bts591 (cited on p. 49).

[176] R. Zhang, M. J. Cairelli, M. Fiszman, *et al.*, "Using semantic predications to uncover drug–drug interactions in clinical data," *Journal of Biomedical Informatics*, vol. 49, pp. 134–147, Jun. 2014, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2014.01.004 (cited on pp. 49, 52).

[177] R. Zhang, D. Hristovski, D. Schutte, A. Kastrin, M. Fiszman, and H. Kilicoglu, "Drug repurposing for covid-19 via knowledge graph completion," en, *Journal of Biomedical Informatics*, vol. 115, p. 103 696, Mar. 2021, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2021.103696 (cited on pp. 51, 52).

[178] N. Konstantinova, S. C. de Sousa, N. P. Cruz, M. J. Maña, M. Taboada, and R. Mitkov, "A review corpus annotated for negation, speculation and their scope," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey: European Language Resources Association (ELRA), May 2012, pp. 3190–3195. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/533_Paper.pdf (cited on pp. 51, 67, 72).

[179] Y. Gu, R. Tinn, H. Cheng, *et al.*, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare*, vol. 3, no. 1, 2:1–2:23, Oct. 2021, ISSN: 2691-1957. DOI: 10.1145/3458754 (cited on p. 52).

[180] C. Zerva, R. Batista-Navarro, P. Day, and S. Ananiadou, "Using uncertainty to link and rank evidence from biomedical literature for model curation," *Bioinformatics*, vol. 33, no. 23, pp. 3784–3792, Dec. 2017, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx466 (cited on p. 53).

[181] A. Morales-Sánchez, T. Molina-Muñoz, J. L. E. Martínez-López, *et al.*, "No association between epstein-barr virus and mouse mammary tumor virus with breast cancer in mexican women," eng, *Scientific Reports*, vol. 3, p. 2970, Oct. 2013, ISSN: 2045-2322. DOI: 10.1038/srep02970 (cited on p. 60).

[182] J. S. Lawson, W. H. Günzburg, and N. J. Whitaker, "Viruses and human breast cancer," eng, *Future Microbiology*, vol. 1, no. 1, pp. 33–51, Jun. 2006, ISSN: 1746-0921. DOI: 10.2217/17460913.1.1.33 (cited on p. 60).

[183] R. Cann, "3.3.1 negation," in *Formal Semantics: An Introduction* (Cambridge Textbooks in Linguistics), Cambridge Textbooks in Linguistics. Cambridge University Press, 1993, pp. 60–61. DOI: 10.1017/CBO9781139166317.004 (cited on p. 64).

[184] R. Cann, "9.3.1 simple modality," in *Formal Semantics: An Introduction* (Cambridge Textbooks in Linguistics), Cambridge Textbooks in Linguistics. Cambridge University Press, 1993, pp. 270–276. DOI: 10.1017/CBO9781139166317.010 (cited on p. 64).

[185] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2013.50 (cited on pp. 64, 65).

[186] V. Balasubramanian, I. Kobyzev, H. Bahuleyan, I. Shapiro, and O. Vechtomova, "Polarized-VAE: Proximity based disentangled representation learning for text generation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online: Association for Computational Linguistics, Apr. 2021, pp. 416–423. DOI: 10.18653/v1/2021.eacl-main.32. [Online]. Available: https://aclanthology.org/2021.eacl-main.32 (cited on pp. 64, 67, 71, 75).

[187] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 424–434. DOI: 10.18653/v1/P19-1041. [Online]. Available: https://www.aclweb.org/anthology/P19-1041 (cited on pp. 64, 67, 70, 71, 73).

[188] P. Cheng, M. R. Min, D. Shen, *et al.*, "Improving disentangled text representation learning with information-theoretic guidance," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 7530–7541. DOI: 10.18653/v1/2020.

acl-main.673. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.673 (cited on pp. 64, 67, 70, 71, 73, 75).

[189] Y. Bao, H. Zhou, S. Huang, *et al.*, "Generating sentences from disentangled syntactic and semantic spaces," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6008–6019. DOI: 10.18653/v1/P19-1602. [Online]. Available: https://www.aclweb.org/anthology/P19-1602 (cited on pp. 64, 67, 75).

[190] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 1587–1596. [Online]. Available: http://proceedings.mlr.press/v70/hu17e.html (cited on pp. 64, 67, 73, 91).

[191] P. Colombo, P. Piantanida, and C. Clavel, "A novel estimator of mutual information for learning to disentangle textual representations," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online: Association for Computational Linguistics, Aug. 2021, pp. 6539–6550. DOI: 10.18653/v1/2021.acl-long.511. [Online]. Available: https://aclanthology.org/2021.acl-long.511 (cited on pp. 64, 71).

[192] J. Vasilakes, C. Zerva, M. Miwa, and S. Ananiadou, "Learning disentangled representations of negation and uncertainty," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8380–8397. DOI: 10.18653/v1/2022.acl-long.574. [Online]. Available: https://aclanthology.org/2022.acl-long.574 (cited on p. 64).

[193] A. Ganguly and S. W. F. Earp, "An introduction to variational inference," no. arXiv:2108.13083, Nov. 2021, arXiv:2108.13083 [cs, stat]. DOI: 10.48550/arXiv.2108.13083. [Online]. Available: http://arxiv.org/abs/2108.13083 (cited on p. 65).

[194] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6114 (cited on pp. 65–67).

[195] I. Higgins, L. Matthey, A. Pal, *et al.*, "Beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning*

*Representations*, 2016. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl (cited on pp. 66, 67, 70).

[196] I. Higgins, D. Amos, D. Pfau, *et al.*, "Towards a Definition of Disentangled Representations," *arXiv:1812.02230 [cs, stat]*, Dec. 2018, arXiv: 1812.02230. [Online]. Available: http://arxiv.org/abs/1812.02230 (visited on 02/02/2021) (cited on pp. 66, 70).

[197] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole, "Weakly supervised disentanglement with guarantees," in *International Conference on Learning Representations*, Sep. 2019. [Online]. Available: https://openreview.net/forum?id=HJgSwyBKvr (cited on pp. 66, 70, 75).

[198] C. Eastwood and C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations," in *International Conference on Learning Representations*, Feb. 2018. [Online]. Available: https://openreview.net/forum?id=By-7dz-AZ (cited on pp. 67, 70, 74).

[199] J. Chen and K. Batmanghelich, "Weakly supervised disentanglement by pairwise similarities," *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 3495–3502, Feb. 2020, ISSN: 2159-5399. DOI: 10.1609/aaai.v34i04.5754 (cited on p. 67).

[200] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 10–21. DOI: 10.18653/v1/K16-1002. [Online]. Available: https://aclanthology.org/K16-1002 (cited on p. 67).

[201] R. Farkas, V. Vincze, G. Móra, J. Csirik, and G. Szarvas, "The conll-2010 shared task: Learning to detect hedges and their scope in natural language text," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning—Shared Task*, 2010, pp. 1–12. [Online]. Available: https://aclanthology.org/W10-3001/ (cited on p. 67).

[202] V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik, "The bioscope corpus: Biomedical texts annotated for uncertainty, negation and their scopes," *BMC Bioinform.*, vol. 9, no. S-11, 2008. DOI: 10.1186/1471-2105-9-S11-S9. [Online]. Available: https://doi.org/10.1186/1471-2105-9-S11-S9 (cited on p. 67).

[203] S. M. Jiménez-Zafra, M. Taulé, M. T. Martín-Valdivia, L. A. Urena-López, and M. A. Martí, "SFU review SP-NEG: A Spanish corpus annotated with negation for sentiment analysis. a typology of negation patterns," *Language Resources and*

*Evaluation*, vol. 52, no. 2, pp. 533–569, 2018. [Online]. Available: https://link.springer.com/article/10.1007/s10579-017-9391-x (cited on p. 67).

[204] P. Cheng, W. Hao, S. Dai, J. Liu, Z. Gan, and L. Carin, "CLUB: A Contrastive Log-ratio Upper Bound of mutual information," in *International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 1779–1788. [Online]. Available: http://proceedings.mlr.press/v119/cheng20b.html (cited on pp. 71, 72, 75).

[205] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, "Style transfer in text: Exploration and evaluation," in *AAAI Conference on Artificial Intelligence*, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17015 (cited on p. 71).

[206] V. L. Rubin, "Stating with certainty or stating with doubt: Intercoder reliability results for manual annotation of epistemically modalized statements," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, Rochester, New York: Association for Computational Linguistics, Apr. 2007, pp. 141–144. [Online]. Available: https://aclanthology.org/N07-2036 (cited on p. 73).

[207] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2078195 (cited on pp. 73, 74).

[208] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in VAEs," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18, Curran Associates Inc., Dec. 2018, pp. 2615–2625. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3327144.3327186 (cited on p. 74).

[209] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLOS ONE*, vol. 9, no. 2, e87357, Feb. 2014, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0087357 (cited on p. 74).

[210] A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in PyTorch," 2017. [Online]. Available: https://openreview.net/forum?id=BJJsrmfCZ (cited on p. 75).

[211] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980 (cited on p. 75).

[212] T. Bosc and P. Vincent, "Do sequence-to-sequence VAEs learn global features of sentences?" In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Nov. 2020, pp. 4296–4318. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.350 (cited on pp. 79, 82).

[213] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," *arXiv:1901.05534 [cs, stat]*, Jan. 2019, arXiv: 1901.05534. [Online]. Available: http://arxiv.org/abs/1901.05534 (cited on p. 84).

[214] K. Yang, T. Zhang, and S. Ananiadou, "Disentangled variational autoencoder for emotion recognition in conversations," *IEEE Transactions on Affective Computing*, pp. 1–12, 2023, ISSN: 1949-3045. DOI: 10.1109/TAFFC.2023.3280038 (cited on p. 84).

[215] P. Kumaraswamy, "A generalized probability density function for double-bounded random processes," en, *Journal of Hydrology*, vol. 46, no. 1, pp. 79–88, Mar. 1980, ISSN: 0022-1694. DOI: 10.1016/0022-1694(80)90036-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022169480900360 (cited on pp. 85, 164).

[216] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee (cited on p. 85).

[217] M. Figurnov, S. Mohamed, and A. Mnih, "Implicit reparameterization gradients," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/92c8c96e4c37100777c7190b76d28233-Abstract.html (cited on p. 85).

[218] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman, *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*, version 1.0, If you use this software, please cite it using these metadata., Mar. 2021. DOI: 10.5281/zenodo.5297715. [Online]. Available: https://doi.org/10.5281/zenodo.5297715 (cited on p. 90).

[219] J. Vasilakes, P. Georgiadis, N. T. H. Nguyen, M. Miwa, and S. Ananiadou, "Contextualized medication event extraction with levitated markers," *Journal of Biomedical*

*Informatics*, vol. 141, p. 104 347, May 2023, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2023.104347 (cited on p. 96).

[220] X. Li, J. Shen, X. Gao, and X. Wang, "Exploiting rich features for detecting hedges and their scope," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, R. Farkas, V. Vincze, G. Szarvas, G. Móra, and J. Csirik, Eds., Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 78–83. [Online]. Available: https://aclanthology.org/W10-3011 (cited on p. 98).

[221] H. Kilicoglu and S. Bergler, "Recognizing speculative language in biomedical research articles: A linguistically motivated perspective," en, *BMC Bioinformatics*, vol. 9, no. 11, S10, Nov. 2008, ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-S11-S10 (cited on p. 98).

[222] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 107–117. DOI: 10.18653/v1/D16-1011. [Online]. Available: https://aclanthology.org/D16-1011 (cited on p. 98).

[223] J. Bastings, W. Aziz, and I. Titov, "Interpretable neural predictions with differentiable binary variables," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2963–2977. DOI: 10.18653/v1/P19-1284. [Online]. Available: https://aclanthology.org/P19-1284 (cited on p. 98).

[224] V. Niculae and M. Blondel, "A regularized framework for sparse and structured neural attention," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/2d1b2a5ff364606ff041650887723470-Abstract.html (cited on p. 98).

[225] A. Bibal, R. Cardon, D. Alfter, *et al.*, "Is attention explanation? an introduction to the debate," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3889–3900. DOI: 10.18653/v1/2022.acl-long.269. [Online]. Available: https://aclanthology.org/2022.acl-long.269 (cited on p. 98).

[226] C. Meister, S. Lazov, I. Augenstein, and R. Cotterell, "Is sparse attention more interpretable?" In *Proceedings of the 59th Annual Meeting of the Association for*

*Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 122–129. DOI: 10.18653/v1/2021.acl-short.17. [Online]. Available: https://aclanthology.org/2021.acl-short.17 (cited on p. 98).

[227] B. Bai, J. Liang, G. Zhang, H. Li, K. Bai, and F. Wang, "Why attentions may not be interpretable?" In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery  Data Mining*, ser. KDD '21, New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 25–34, ISBN: 978-1-4503-8332-5. DOI: 10.1145/3447548.3467307. [Online]. Available: https://dl.acm.org/doi/10.1145/3447548.3467307 (cited on p. 98).

[228] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, T. Linzen, G. Chrupała, Y. Belinkov, and D. Hupkes, Eds., Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. DOI: 10.18653/v1/W19-4828. [Online]. Available: https://aclanthology.org/W19-4828 (cited on p. 99).

[229] S. Vashishth, S. Upadhyay, G. S. Tomar, and M. Faruqui, "Attention interpretability across nlp tasks," *arXiv preprint arXiv:1909.11218*, no. arXiv:1909.11218, Sep. 2019, arXiv:1909.11218. DOI: 10.48550/arXiv.1909.11218. [Online]. Available: http://arxiv.org/abs/1909.11218 (cited on p. 99).

[230] M. Neumann, D. King, I. Beltagy, and W. Ammar, "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 319–327. DOI: 10.18653/v1/W19-5034. eprint: arXiv:1902.07669. [Online]. Available: https://www.aclweb.org/anthology/W19-5034 (cited on p. 101).

[231] A. Laha, S. A. Chemmengath, P. Agrawal, M. Khapra, K. Sankaranarayanan, and H. G. Ramaswamy, "On controllable sparse alternatives to softmax," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018, pp. 6423–6433 (cited on p. 102).

[232] B. Cui, Y. Li, M. Chen, and Z. Zhang, "Fine-tune bert with sparse self-attention mechanism," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Com-

putational Linguistics, Nov. 2019, pp. 3548–3553. DOI: 10.18653/v1/D19-1361. [Online]. Available: https://aclanthology.org/D19-1361 (cited on p. 102).

[233] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019, pp. 8026–8037. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf (cited on p. 106).

[234] A. C. Stickland and I. Murray, "Bert and pals: Projected attention layers for efficient adaptation in multi-task learning," en, in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 5986–5995. [Online]. Available: https://proceedings.mlr.press/v97/stickland19a.html (cited on pp. 111, 112, 121).

[235] J.-D. Kim, T. Ohta, and J. Tsujii, "Corpus annotation for mining biomedical events from literature," *BMC Bioinformatics*, vol. 9, no. 1, p. 10, Jan. 2008, ISSN: 1471-2105. DOI: 10.1186/1471-2105-9-10 (cited on p. 115).

[236] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," en, in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 3319–3328. [Online]. Available: https://proceedings.mlr.press/v70/sundararajan17a.html (cited on p. 116).

[237] J. Enguehard, "Sequential integrated gradients: A simple but effective method for explaining language models," en, in *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada: Association for Computational Linguistics, 2023, pp. 7555–7565. DOI: 10.18653/v1/2023.findings-acl.477. [Online]. Available: https://aclanthology.org/2023.findings-acl.477 (cited on p. 116).

[238] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," *arXiv preprint*, no. arXiv:1612.08220, Jan. 2017, arXiv:1612.08220. DOI: 10.48550/arXiv.1612.08220. [Online]. Available: http://arxiv.org/abs/1612.08220 (cited on p. 116).

[239] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," en, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 1135–1144, ISBN: 978-1-4503-4232-2. DOI:

10.1145/2939672.2939778. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939778 (cited on p. 116).

[240] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (cited on p. 116).

[241] J. DeYoung, S. Jain, N. F. Rajani, *et al.*, "Eraser: A benchmark to evaluate rationalized nlp models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 4443–4458. DOI: 10.18653/v1/2020.acl-main.408. [Online]. Available: https://aclanthology.org/2020.acl-main.408 (cited on pp. 116, 120).

[242] A. Üstün, V. Aryabumi, Z.-X. Yong, *et al.*, "Aya model: An instruction finetuned open-access multilingual language model," no. arXiv:2402.07827, Feb. 2024, arXiv:2402.07827 [cs]. DOI: 10.48550/arXiv.2402.07827. [Online]. Available: http://arxiv.org/abs/2402.07827 (cited on p. 120).

[243] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020 (cited on p. 121).

[244] OpenAI, J. Achiam, S. Adler, *et al.*, "Gpt-4 technical report," no. arXiv:2303.08774, Mar. 2024, arXiv:2303.08774 [cs]. DOI: 10.48550/arXiv.2303.08774. [Online]. Available: http://arxiv.org/abs/2303.08774 (cited on p. 121).

[245] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023 (cited on p. 121).

[246] F. Petroni, T. Rocktäschel, S. Riedel, *et al.*, "Language models as knowledge bases?" In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. [Online]. Available: https://www.aclweb.org/anthology/D19-1250 (cited on p. 121).

[247] I. Augenstein, T. Baldwin, M. Cha, *et al.*, "Factuality challenges in the era of large language models," no. arXiv:2310.05189, Oct. 2023, arXiv:2310.05189 [cs]. DOI: 10.48550/arXiv.2310.05189. [Online]. Available: http://arxiv.org/abs/2310.05189 (cited on p. 121).

[248] *Ai-generated answers temporarily banned on coding qa site stack overflow - the verge*. [Online]. Available: https://www.theverge.com/2022/12/5/23493932/chatgpt-ai-generated-answers-temporarily-banned-stack-overflow-llms-dangers (cited on p. 121).

[249] *Can chatgpt fact-check? we tested. - poynter*. [Online]. Available: https://www.poynter.org/fact-checking/2023/chatgpt-ai-replace-fact-checking/ (cited on p. 121).

[250] S. Yang, E. Gribovskaya, N. Kassner, M. Geva, and S. Riedel, "Do large language models latently perform multi-hop reasoning?" *arXiv preprint arXiv:2402.16837*, 2024 (cited on p. 121).

[251] Y. Gal, "Uncertainty in deep learning," en, Ph.D. dissertation, University of Cambridge, Sep. 2016. [Online]. Available: http://mlg.eng.cam.ac.uk/yarin/thesis/thesis.pdf (cited on p. 161).

[252] S. Tratz and E. Hovy, "A taxonomy, dataset, and classifier for automatic noun compound interpretation," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010, pp. 678–687. [Online]. Available: https://aclanthology.org/P10-1070 (cited on pp. 161, 165).

[253] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning whom to trust with MACE," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 1120–1130. [Online]. Available: https://aclanthology.org/N13-1132 (cited on pp. 161, 163, 169).

[254] K. L. Gwet, "Computing inter-rater reliability and its variance in the presence of high agreement," en, *British Journal of Mathematical and Statistical Psychology*, vol. 61, no. 1, pp. 29–48, 2008, ISSN: 2044-8317. DOI: 10.1348/000711006X126600 (cited on p. 161).

[255] S. Jagabathula, L. Subramanian, and A. Venkataraman, "Identifying unreliable and adversarial workers in crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 18, no. 93, pp. 1–67, 2017, ISSN: 1533-7928 (cited on p. 161).

[256] M. Li, A. Fahlström Myrman, T. Mu, and S. Ananiadou, "Modelling instance-level annotator reliability for natural language labelling tasks," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*,

Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2873–2883. DOI: 10.18653/v1/N19-1295. [Online]. Available: https://aclanthology.org/N19-1295 (cited on pp. 161, 163).

[257] Q. Nguyen, H. Valizadegan, and M. Hauskrecht, "Learning classification models with soft-label information," *Journal of the American Medical Informatics Association*, vol. 21, no. 3, pp. 501–508, May 2014, ISSN: 1067-5027. DOI: 10.1136/amiajnl-2013-001964 (cited on p. 161).

[258] K. M. Collins, U. Bhatt, and A. Weller, "Eliciting and learning with soft labels from every annotator," no. arXiv:2207.00810, Aug. 2022, arXiv:2207.00810 [cs]. DOI: 10.48550/arXiv.2207.00810. [Online]. Available: http://arxiv.org/abs/2207.00810 (cited on pp. 161, 169).

[259] B. Plank, D. Hovy, and A. Søgaard, "Linguistically debatable or just plain wrong?" In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 507–511. DOI: 10.3115/v1/P14-2083. [Online]. Available: https://aclanthology.org/P14-2083 (cited on p. 162).

[260] A. N. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, and M. Poesio, "Learning from disagreement: A survey," en, *Journal of Artificial Intelligence Research*, vol. 72, pp. 1385–1470, Dec. 2021, ISSN: 1076-9757. DOI: 10.1613/jair.1.12752 (cited on pp. 162, 165, 169).

[261] J. Baan, W. Aziz, B. Plank, and R. Fernández, "Stop measuring calibration when humans disagree," no. arXiv:2210.16133, Nov. 2022, arXiv:2210.16133 [cs]. DOI: 10.48550/arXiv.2210.16133. [Online]. Available: http://arxiv.org/abs/2210.16133 (cited on p. 162).

[262] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991, ISSN: 1557-9654. DOI: 10.1109/18.61115 (cited on p. 165).

[263] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," en, *Advances in Neural Information Processing Systems*, vol. 31, 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/3ea2db50e62ceefceaf70a Abstract.html (cited on pp. 167, 168).

[264] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," en, *arXiv:1806.01768 [cs, stat]*, Oct. 2018, arXiv: 1806.01768. [Online]. Available: http://arxiv.org/abs/1806.01768 (cited on p. 167).

[265] T. Joo, U. Chung, and M.-G. Seo, "Being bayesian about categorical probability," en, in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 4950–4961. [Online]. Available: https://proceedings.mlr.press/v119/joo20a.html (cited on p. 167).

[266] J. C. Peterson, R. M. Battleday, T. L. Griffiths, and O. Russakovsky, "Human uncertainty makes classification more robust," 2019, pp. 9617–9626. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Peterson_Human_Uncertainty_Makes_Classification_More_Robust_ICCV_2019_paper.html (cited on pp. 168, 169).

[267] A. N. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, and M. Poesio, "A case for soft loss functions," en, *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 8, pp. 173–177, Oct. 2020. [Online]. Available: https://ojs.aaai.org/index.php/HCOMP/article/view/7478 (cited on pp. 168, 169).

[268] A. Malinin and M. Gales, "Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/7dd2ae7db7d18ee7c9425e38d
Abstract.html (cited on p. 168).

[269] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012 (cited on p. 169).

[270] A. Dumitrache, L. Aroyo, and C. Welty, "Crowdsourcing ground truth for medical relation extraction," *ACM Transactions on Interactive Intelligent Systems*, vol. 8, no. 2, 11:1–11:20, Jul. 2018, ISSN: 2160-6455. DOI: 10.1145/3152889 (cited on p. 169).

[271] A. Cercas Curry, G. Abercrombie, and V. Rieser, "Convabuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational ai," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7388–7403. DOI: 10.18653/v1/2021.emnlp-main.587. [Online]. Available: https://aclanthology.org/2021.emnlp-main.587 (cited on p. 169).

[272] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, pp. 770–778. [Online]. Available: https://openaccess.thecvf.com/

content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html (cited on p. 169).

[273] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423 (cited on p. 169).

[274] A. Cercas Curry, G. Abercrombie, and V. Rieser, "ConvAbuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational AI," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7388–7403. DOI: 10.18653/v1/2021.emnlp-main.587. [Online]. Available: https://aclanthology.org/2021.emnlp-main.587 (cited on p. 169).

[275] A. Dumitrache, O. Inel, L. Aroyo, B. Timmermans, and C. Welty, "Crowdtruth 2.0: Quality metrics for crowdsourcing with disagreement," in *SAD/CrowdBias@HCOMP*, Aug. 2018. DOI: 10.48550/arXiv.1808.06080. [Online]. Available: http://arxiv.org/abs/1808.06080 (cited on p. 170).

[276] T. Fornaciari, A. Uma, S. Paun, B. Plank, D. Hovy, and M. Poesio, "Beyond black white: Leveraging annotator disagreement via soft-label multi-task learning," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 2591–2597. DOI: 10.18653/v1/2021.naacl-main.204. [Online]. Available: https://aclanthology.org/2021.naacl-main.204 (cited on p. 170).

[277] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "Brat: A web-based tool for nlp-assisted text annotation," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 102–107 (cited on p. 172).

# Appendices

# Appendix A

# Subjective Logic Encodings

Section 4.7.5 briefly discussed the possibility of utilizing Subjective Logic for representing second order uncertainty regarding data annotations. This section expands upon this idea, introducing Subjective Logic Encodings (SLEs) as a method for representing and learning from distributions over labels and providing preliminary experiments regarding their usefulness.

## A.1 Sources of Uncertainty in Annotation

We are interested in modeling *aleatoric uncertainty*, also called data uncertainty, which refers to the irreducible uncertainty inherent in the data or events being modeled. It stands in contrast to *epistemic uncertainty* —due to a lack of knowledge regarding the events— and *distributional uncertainty* —uncertainty regarding out-of-distribution (OOD) events. Generally, aleatoric uncertainty arises from noise inherent in the event being modeled (e.g., a dice roll) or from measurement noise [251]. More specifically, we view annotation as a measurement process and break it down into three indicators of uncertainty: inter-annotator disagreement indicates inherent event uncertainty, while annotator reliability and confidence indicate measurement uncertainty. We discuss each of these indicators below.

**Annotator Reliability:** An annotator may be highly confident in their annotations but still often provide incorrect or inconsistent labels due to a lack of expertise, the difficulty of the annotation task, etc. Conversely, adversarial annotators may purposefully provide poor annotations. Annotator reliability is thus an indicator of measurement noise. Simple measures of reliability are average agreement between annotators [252], average Cohen's $\kappa$ between pairs of annotators [253], and the G-index [254]. Previous works have also learned models of annotator reliability directly from data [253], [255], [256].

**Annotator Confidence:** Borderline examples, difficult annotation tasks, or unclear annotation guidelines can all influence an annotator's uncertainty in their assigned labels. Thus individual annotator confidence is an indicator of measurement noise. Previous works have measured confidence by asking annotators additional questions [257] or eliciting probabilistic labels [258].

**Inter-Annotator Disagreement:** Even assuming perfect annotator reliability and confidence, multiple annotators may still assign different labels to the same event, indicating an inherent uncertainty in the event (or again the annotation task/guidelines) [259]. Inter-annotator disagreement is thus an indicator of aleatoric uncertainty. There has been a recent surge of interest in leveraging disagreements between annotators for improving classification (see [260] for a survey).

It is commonly assumed that crowd annotations are samples from the same categorical distribution over labels for a given example (e.g., this assumption is explicitly stated in [261]). The existence of annotator confidence and reliability, however, suggest that such samples pass through a intermediate "measurement" process, i.e., the subjective opinion of the annotator, which lends uncertainty to each annotation. It is therefore necessary to explicitly encode annotations as opinions —i.e., including information regarding annotator confidence and reliability.

## A.2 Background: Subjective Logic Theory

While the above shows that it is natural to view annotations as subjective opinions of the annotators, the additional uncertainty dimensions are incompatible with the current standard vector representations which are limited to probabilities. Subjective Logic (SL), introduced in Section 3.4.1, is a type of probabilistic logic that explicitly encodes opinions of agents regarding a domain of possible events and can be used to encode these additional dimensions. We here provide a brief overview of the aspects of SL necessary to understand SLEs. This differs from the overview given in Section 3.4.1 in that the definitions assume Dirichlet distributions in place of Beta distributions. Whether we use Beta or Dirichlet distributions ultimately matters little (and depends on the dimensionality of the labels), but Dirichlets result in more generic definitions.

#### Opinion Representation

In SL, a subjective opinion regarding some item $i$ according to an agent $m$ over a domain of $K$ possible events (e.g., class labels) is denoted $\omega_m^{(i)} = (\boldsymbol{b}_m^{(i)}, u_m^{(i)}, \boldsymbol{a})$, where $\boldsymbol{b}_m^{(i)} \in [0,1]^K$ is a vector of beliefs, $u_m^{(i)} \in [0,1]$ represents the uncertainty of the opinion, and $\boldsymbol{a} \in [0,1]^K$ is a vector of base rates or prior probabilities over the event space[1]. These parameters are subject to the constraint $u_m^{(i)} + \sum_{k=1}^{K} b_{m,k}^{(i)} = 1$. SL opinions can be reparameterized as Dirichlet distributions, and the mapping to Dirichlet parameters $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha}_m^{(i)} = \frac{2\boldsymbol{b}_m^{(i)}}{u_m^{(i)}} + K\boldsymbol{a} \tag{A.1}$$

[1]Without loss of generality, we assume uniform priors throughout this work.

**Combining Opinions**

A key feature of SL opinions is that they may be combined to form consensus opinions using various operators. In this work, we utilize the cumulative belief fusion and trust discounting operators. Cumulative belief fusion, denoted $\bigoplus$ and introduced already in Equation (3.4) of the main text, combines opinions regarding a single event —such as two annotators observing a single training example— treating each as evidence of the true label distribution. Cumulative fusion reduces the uncertainty $u$ by combining evidence. This means that as more opinions are fused, the uncertainty tends towards zero and the resulting Dirichlet approaches zero variance, equivalent to a categorical probability.

The trust discounting operator increases the uncertainty of an opinion according to a separate opinion of the reliability of that annotator. It is denoted $\bigotimes$ and defined as

$$\hat{\omega}_m^{(i)} = \omega_R^{(m)} \bigotimes \omega_m^{(i)} = \begin{cases} \hat{\boldsymbol{b}}_m^{(i)} & = \boldsymbol{P}_R^{(m)} \boldsymbol{b}_m^{(i)} \\ \hat{u}_m^{(i)} & = 1 - \boldsymbol{P}_R^{(m)} \sum_{k=1}^{K} b_{m,k}^{(i)} \\ \hat{\boldsymbol{a}}_m & = \boldsymbol{a}_m \end{cases} \tag{A.2}$$

where $\omega_R^{(m)}$ is an opinion of the reliability of the subjective opinion $\omega_m^{(i)}$.

## A.3 Constructing SLEs

We can utilize cumulative fusion and trust discounting to encode and aggregate labels as SLEs. Let there be a dataset of $N$ annotated examples $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^{N} \in (\mathcal{X}, \mathcal{Y})^N$, where $\mathcal{X} \in \mathbb{R}^d$ is the input space and $\mathcal{Y} \in \{1, ..., K\}^M$ is the label space over $K$ labels and $M$ annotators. That is, each $\boldsymbol{y}^{(i)}$ is a vector of class labels from all $M$ annotators[2]. Further, each individual judgment $y_m^{(i)}$ has metadata regarding the annotator's reliability $r_m^{(i)}$ and subjective uncertainty $u_m^{(i)}$[3]. Given this, our goal for constructing the target distributions is twofold:

1. Encode each individual judgement taking into account each of the sources of uncertainty described in Section A.1.
2. Define a method for aggregating individual encoded judgments into a target distribution for training a classification model that accounts for label disagreement.

As discussed in Section A.2, the opinion of annotator $m$ regarding the $i^{th}$ example is $\omega_m^{(i)} = (\boldsymbol{b}_m^{(i)}, u_m^{(i)}, \boldsymbol{a}_m)$. In practice, $u_m^{(i)}$ can be any mapping from a user-supplied indicator of subjective uncertainty to the range $[0, 1]$. The belief vector $\boldsymbol{b}_m^{(i)}$ is computed using the subjective uncertainty as

[2]For simplicity, we assume each example has the same number of annotators, but we note that this is not a requirement for label aggregation.

[3]Our approach is agnostic to whether reliability is an overall score for the annotator [253] or an instance level score from another model [256]

$$\boldsymbol{b}_m^{(i)} = \tilde{\boldsymbol{y}}_m^{(i)} - u_m^{(i)} \tilde{\boldsymbol{y}}_m^{(i)} \tag{A.3}$$

where $\tilde{\boldsymbol{y}}_m^{(i)}$ is the vector encoding of $y_m^{(i)}$, such as one-hot or probabilities[4].

Finally, we compute the SLE as the aggregate of the opinions for a given example $x$ using the cumulative fusion Equation (3.4) and trust discounting operators Equation (A.4)

$$\omega_\Diamond^{(i)} = \bigoplus_{m=1}^{M} \omega_R^{(m)} \bigotimes \omega_m^{(i)} \tag{A.4}$$

In the case where there is no information regarding reliability ($r_m^{(i)} = 1$) or uncertainty ($u_m^{(i)} = 0$), this process results in a dogmatic opinion $\omega_\Diamond^{(i)} = (\boldsymbol{b}_\Diamond^{(i)}, 0, \mathbf{1}/K)$ which is equivalent to a categorical probability. In the further case where there is no disagreement between annotators, the resulting opinion is equivalent to a one-hot encoded target.

## A.4 Synthetic Data Experiments

To our knowledge, there are no real-world datasets that contain all the sources of uncertainty discussed in this work. We therefore illustrate the benefit of using SLEs in place of other methods for label aggregation using synthetic data. We generate synthetic datasets to assess the ability of SLEs to recover the true label from a set of crowd annotations that have been corrupted according to a range of annotator reliabilities and confidences. We conduct three experiments: (1) assuming perfect certainty, we vary total annotator reliability from very low to perfect; (2) assuming high reliability, we vary annotator confidence from very low to perfect; (3) as for (2), but we assume low annotator reliability.

### A.4.1 Data Generation

We generate a set of $N$ 5-class true labels as evenly spaced points on the simplex. For each true label, we generate crowd annotations from $M = 10$ annotators with varying degrees of confidence and reliability. Specifically, we draw values of these parameters from Beta distributions and use them to corrupt the true labels: a true label is permuted with probability equal to reliability and it is recalibrated according to confidence. This process is detailed in Algorithm 1 below, where $\text{permute}(\boldsymbol{y}, r)$ shuffles the indices of $\boldsymbol{y}$ with probability $r$ and $\text{recalibrate}(\boldsymbol{y}, c) = \frac{\exp(\ln \boldsymbol{y}_j c)}{\sum_{j=1}^{d} \exp(\ln \boldsymbol{y}_j c)}$ smooths the probabilities in $\boldsymbol{y}$ according to the confidence $c$. This process is an identity when $c = 1$ and $r = 1$ and pushes $\boldsymbol{y}$ towards the uniform distribution as $c \to 0$. Annotator certainty and reliability can thus be varied by specifying different $\alpha$ and $\beta$ parameters to the beta distributions.

Using this process we generate synthetic datasets in three different scenarios according to

[4]For binary labels ($K = 2$), the distribution is a Beta with $\alpha$ or $\beta = 1$, equivalent to a Kumaraswamy distribution [215].

**Algorithm 1** Synthetic Data Generation Process

**Require:** Two sets of Beta distribution parameters $(\alpha, \beta)$ and $(\alpha', \beta')$

**for** $m \in M$ **do**
  $c \sim Beta(\alpha, \beta)$ ▷ Sample annotator confidence.
  $r \sim Beta(\alpha', \beta')$ ▷ Sample annotator reliability.
  **for** $\boldsymbol{y} \in \triangle^{(d-1)}$ **do**
    $\boldsymbol{y}^* \leftarrow \text{permute}(\boldsymbol{y}, r)$ ▷ Permute $\boldsymbol{y}$ with probability $r$.
    $\boldsymbol{y}^* \leftarrow \text{recalibrate}(\boldsymbol{y}^*, c)$ ▷ Recalibrate probabilities.
  **end for**
**end for**

different values of the $\alpha$ and $\beta$ parameters. In the first, we assume no information regarding annotator confidence ($c = 1$) and only vary the reliability of the annotators (Figure A.1a). In the second, we assume high reliability and vary annotator confidence (Figure A.1b). In the third, we assume low reliability and vary annotator confidence (Figure A.1c).

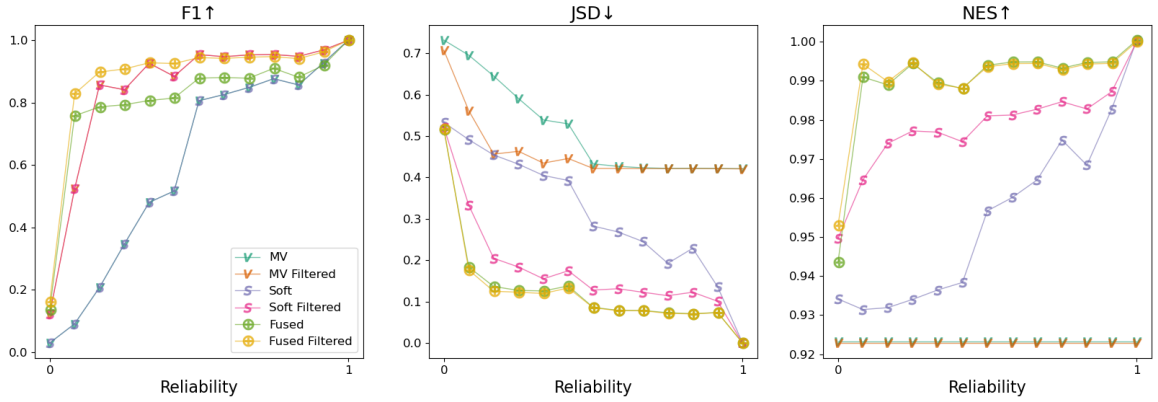
### A.4.2 Evaluation

The evaluation uses both hard and soft evaluation metrics between the aggregated crowd annotations and the true labels. For the hard metric we report the micro-averaged F1 on the test set, and label predictions are obtained from SLEs by taking the argmax of the mode of the predicted Dirichlet, i.e., $\hat{y} = \text{argmax}_k \, \mathbb{E}_{Q_\theta}(y_k|\boldsymbol{x})$. Where the F1 measures the ability of the model to make correct predictions, the soft evaluation metrics measure how well the model is able to match the distribution of the crowd annotations. Specifically, we use the Jensen-Shannon Divergence (JSD) [262] and the Normalized Entropy Similarity (NES) [260] between the aggregated and true distributions.

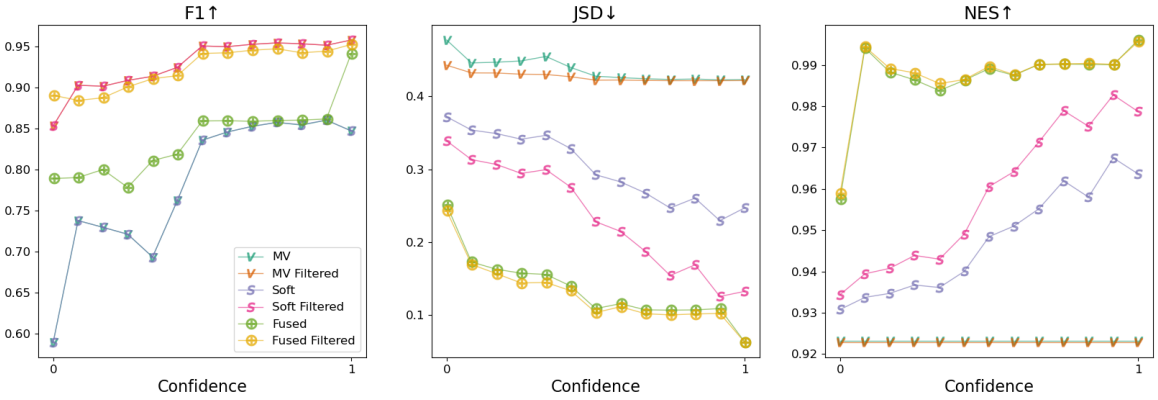
For each experiment, we plot the F1, JSD, and NES curves over the range of uncertainty values. We compare SLEs to two baseline aggregation methods: Majority Voting (MV) chooses the class with the greatest frequency in the crowd annotations, resulting in a one-hot label; Soft Voting (Soft) computes the average count of each label from the crowd annotations, resulting in a vector of continuous labels. We evaluate each aggregation method using all crowd annotations as well as a subset filtered according to reliability as in [252] (Filtered). For all experiments, we average the results over 10 runs with crowd-annotations generated according to different random seeds.

### A.4.3 Results

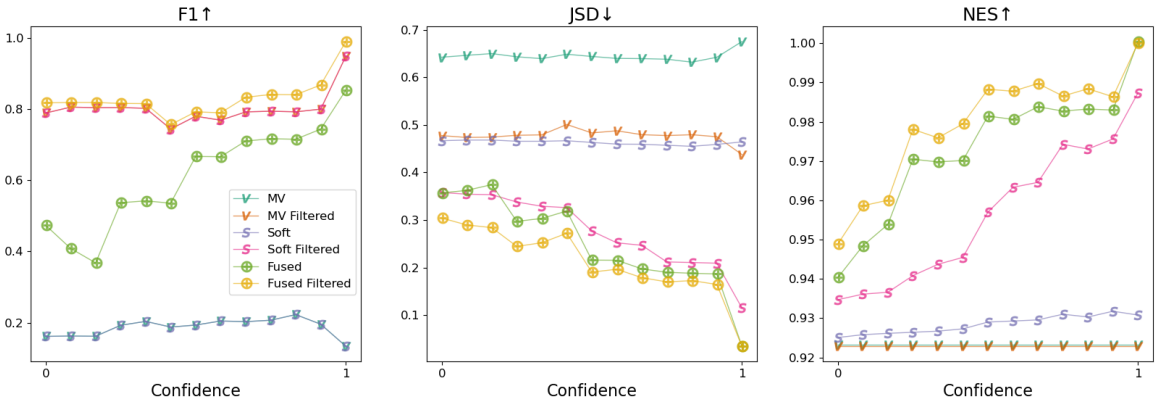
The plots in Figure A.1 show that SLEs are better able to recover the true distribution from crowd annotations than the baselines, measured by JSD and NES. Additionally, the ability of SLEs to "deweight" annotations according to uncertainty means that they achieve higher F1s than the unfiltered baselines. Filtering annotations according to an estimate of annotator reliability improves results for all methods, but we note that the difference in JSD and NES between unfiltered and filtered SLEs is much less than MV and Soft, suggesting the SLEs are better able to capture the true distribution even in the presence of noisy labels.




(a) Varying reliabilities from very low $r \sim Beta(1, 10)$ to perfect $r = 1$, assuming perfect confidence $c = 1$ for all annotators.




(b) Varying confidences from very low $c \sim Beta(1, 10)$ to perfect $c = 1$, assuming high reliability $r \sim Beta(10, 1)$.




(c) Varying confidences from very low $c \sim Beta(1, 10)$ to perfect $c = 1$, assuming low reliability $r \sim Beta(1, 10)$.

Figure A.1. F1, JSD, and NES results for Majority Voting (MV), Soft voting (Soft), and SLE cumulative fusion (Fused) given both all and filtered annotations on the synthetic data across different ranges of uncertainty. Results using CrowdTruth were nearly identical to Soft, and so omitted here for clarity.

## A.5 Learning SLEs

The sources of aleatoric uncertainty described in Section A.1 have been independently studied in previous works. However, the methods used for incorporating them into classification models are quite specialized and generally incompatible with each other. In this section, we describe in detail how SLEs aggregate each of these sources of uncertainty into a single target for model estimation.

### A.5.1 Dirichlet Neural Networks

Using SLEs in machine learning applications requires a classifier to predict Dirichlet distributions. Predicting Dirichlets has been studied extensively in previous work. [263] introduce Prior Networks, which use Dirichlet outputs to model distributional uncertainty given out-of-distribution data. Concurrently, and most similar to our work, [264] propose a Dirichlet neural network for modeling prediction uncertainty that is explicitly influenced by SL. Still, this work focuses on inducing predictive uncertainty of the model given gold-standard labels rather than utilizing aleatoric uncertainty obtained from data. Also, despite being inspired by SL, their model predicts the $\boldsymbol{\alpha}$ parameters of a Dirichlet instead of the corresponding SL opinion parameters, as we propose to do here. [265] develop a Dirichlet latent variable model, trained using variational inference, to construct a prior over the predicted class probabilities given gold labels. In contrast with these previous works, which model distributional uncertainty given gold-standard labels, this work focuses on encoding aleatoric uncertainty obtained as part of the annotation process.

### A.5.2 Model Estimation

As described in Section A.3, we assume a dataset with aggregated SLE target distributions $\mathcal{D} = \{\boldsymbol{x}^{(i)}, \omega_{\Diamond}^{(i)}\}_{i=1}^{N} \in (\mathcal{X}, \Omega)^N$. We aim to estimate a classification model $\mathcal{F}_\Theta : \mathcal{X} \rightarrow \Omega$ parameterized by $\Theta$ which maps from the input space $\mathcal{X}$ to the space of SL opinions $\Omega$ represented by reparameterized Dirichlet distributions. The process of estimating such a classifier amounts to distribution matching between the target $\omega_{\Diamond}^{(i)}$ and approximating SLE opinions $\omega_{\theta}^{(i)}$, which are predicted by a classifier $f_\theta \in \mathcal{F}_\Theta$. In our setup, a neural network $f_\theta$ predicts the $\boldsymbol{b}$ and $u$ parameters of the opinion representation.

$$\boldsymbol{b}_{\theta}^{(i)}, u_{\theta}^{(i)} \leftarrow \text{softmax}(f_\theta(x^{(i)})) \tag{A.5}$$

where softmax enforces the constraint $u_{\theta}^{(i)} + \sum_{k=1}^{K} b_{\theta,k}^{(i)} = 1$.

In the usual classification setup where the targets are gold-standard labels, model estimation often uses the cross entropy loss between the one-hot target labels $\tilde{\boldsymbol{y}}$ and the label probabilities predicted by the classifier, $Q_\theta$.

$$\mathcal{L}_\theta = \sum_{k=1}^{K} \tilde{y}_k \log Q_\theta(\hat{y} = k|x) \tag{A.6}$$

However, given that we are here concerned with target *distributions* rather than labels, it may be clearer to rewrite the cross-entropy loss according to the KL divergence between the target distribution $P$ and predicted distribution $Q$.

$$\mathcal{L}_\theta = D_{KL}(P \,||\, Q_\theta) + H(P) \tag{A.7}$$

Since the entropy term in Equation (A.7) is a constant that depends only on the target distribution, the minimization of the cross-entropy amounts to the minimization of the KL divergence between the target and the predicted distributions[5]. Thus our target loss function is simply the KL divergence,

$$\mathcal{L}_\theta = D_{KL}(P \,||\, Q_\theta) = \int_x P(x) \log \frac{P(x)}{Q_\theta(x)} \tag{A.8}$$

which is essentially the same loss function proposed by [263], without the additional KL term for out-of-distribution data[6].

Unfortunately, there are two issues with this objective: (1) the KL divergence error surface is poorly suited to optimization when the target distributions are sparse "one-hot" distributions corresponding to dogmatic opinions; (2) this "forward" KL divergence penalizes $Q_\theta$ much less where $P$ is very small, which results in a $Q_\theta$ that covers a majority of the space and is a poor predictor.

To overcome (1), we follow [263] and smooth dogmatic target opinions by redistributing a small $\epsilon$ amount of belief mass to the uncertainty parameter, which adds a small amount of density to the other corners of the probability simplex (Equation (A.9)).

$$\omega_m^{(i)} = \begin{cases} \boldsymbol{b}_m^{(i)} & = \boldsymbol{b}_m^{(i)} - \epsilon \\ u_m^{(i)} & = u_m^{(i)} + \epsilon \\ \boldsymbol{a}_m & = \boldsymbol{a}_m \end{cases} \tag{A.9}$$

To overcome (2), we follow [268] and use the *reverse* KL divergence

$$\mathcal{L}_\theta = D_{KL}(Q \,||\, P_\theta) = \int_x Q_\theta(x) \log \frac{Q_\theta(x)}{P(x)} \tag{A.10}$$

which encourages $Q_\theta$ to fit under $P$.

## A.6 Real-World Data Experiments

Despite the lack of real-world datasets that contain all three sources of uncertainty, we can still compare SLEs to existing methods for learning from crowd annotations. We experiment with the following real-world datasets in computer vision and natural language processing that contain crowd-sourced annotations with multiple annotators per example.

[5] In standard classification setups with a single "gold" judgment per sample, the target is one-hot, i.e., a deterministic categorical distribution over the $K$ labels. In this case, the the entropy of the target distribution $H(P)$ is zero and cross-entropy equals the KL-divergence. However, when using soft labels as in [266], [267], the entropy term is $> 0$, meaning that the values of the cross-entropy loss between hard- and soft-label setups are not comparable.

[6] It would be trivial to add support for OOD data using the additional KL term and OOD training data, but this is out of scope for this work.

- CIFAR-10S [258]: Image classification. A variant of the common CIFAR-10 computer vision benchmark dataset [269] annotated by 6 annotators per example. Additionally, CIFAR-10S elicited probabilistic labels from each annotator to capture each annotator's confidence.
- MRE [270]: A dataset of 4,000 sentences in English from PubMed abstracts expressing a *cause* or *treats* relation between two entities. Each example is annotated by 15 annotators.
- ConvAbuse [271]: A collection of 7,000 sentences sampled from different chatbot systems annotated by 8 annotators for a variety of abuse types.
- RTE [253]: 800 sentences pairs annotated for semantic entailment by 10 annotators.

### A.6.1 Models

We utilize the following models for each real-world dataset.

- CIFAR-10S: Following previous work [258], [267], we employ a ResNet-32A model [272] and replicate their evaluation setup.
- MRE: We use the same model and training procedure as [260]. This is a fine-tuned BERT sentence classifier with a single linear output layer [273].
- ConvAbuse: As for MRE, we use a fine-tuned BERT sentence classifier with a single linear output layer as in [274].
- RTE: We employ the model implemented by [260]: The premise and hypothesis are concatenated, encoded by BERT, passed through 3 ReLU-activated linear layers, and a final linear output layer.

### A.6.2 Evaluation

We compare SLEs to a number of baseline methods previously used to handle annotation uncertainty.

- Majority-Voting (MV): the signal is the most common label from the annotator judgements. If more than one label is tied for most common, one of the them is chosen at random. Model estimation uses the cross-entropy loss.
- Cross-Entropy (CE) loss between the predicted distribution and the categorical distribution computed directly from the crowd labels [266].
- Kullback-Leibler divergence (KL) between the predicted distribution and the categorical distribution computed directly from the crowd labels [260].

- CrowdTruth: a method for aggregating labels that weights them according to measures of annotation, annotator, and example reliability [275]. This method is similar to SLEs in what is measured, but is more prescriptive in how these measurements are computed. In fact, the measurements defined by CrowdTruth could be used by SLEs as measures of annotator uncertainty and reliability. Model estimation uses cross-entropy loss.
- Multi-task Learning from Soft Labels (MTLSL): a simple combination of learning from gold labels and categorical probabilities computed from the crowd labels [276]. Model estimation uses cross-entropy for the gold labels and the reverse KL divergence for the probabilistic labels, as we do for SLEs.

A comparison of these methods to SLEs is given in table Table A.1. We note that, of the methods listed above, only CE and KL are able to learn from annotator confidence, but only via probabilistic labels, and only CrowdTruth is able to learn from annotator reliability. It is possible to combine MTLSL with CE/KL by eliciting soft labels from $> 1$ annotator, but this still lacks the ability to learn from annotator reliability. SLEs, on the other hand, are able to incorporate all three sources of annotation uncertainty and do so without large changes to the model architecture. Furthermore, unlike MTLSL, SLEs do not require each example to be annotated by the same number of annotators, nor do they depend on having both gold-standard and crowd annotations available.

| | MV | CE | KL | CrowdTruth | MTLSL | SLE |
|---|---|---|---|---|---|---|
| Annotator Confidence | | ✓ | ✓ | | | ✓ |
| Annotator Reliability | | | | ✓ | | ✓ |
| Crowd Annotations | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table A.1. A comparison of methods for learning from annotation uncertainties.

### A.6.3 Results

We report the performance of SLEs compared to the baselines on the real world datasets in Table A.2. We report results of methods using both gold and crowd annotations.

| | CIFAR-10S | | | MRE | | | ConvAbuse | | | RTE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F1↑ | JSD↓ | NES↑ | F1↑ | JSD↓ | NES↑ | F1↑ | JSD↓ | NES↑ | F1↑ | JSD↓ | NES↑ |
| Gold | 0.645 | 0.331 | **0.390** | **0.840** | 0.202 | 0.686 | | | | 0.613 | 0.346 | 0.569 |
| SLE (gold) | **0.653** | **0.326** | 0.369 | 0.830 | 0.198 | **0.697** | - | - | - | **0.617** | 0.351 | **0.587** |
| MTLSL* | 0.636 | 0.344 | 0.347 | 0.828 | **0.145** | 0.596 | | | | 0.610 | **0.338** | 0.565 |
| MV | 0.645 | **0.336** | 0.398 | 0.749 | 0.133 | 0.660 | 0.886 | 0.084 | 0.153 | 0.605 | 0.346 | 0.571 |
| CE | 0.661 | 0.356 | **0.551** | 0.751 | 0.127 | 0.749 | 0.898 | **0.072** | 0.375 | **0.611** | 0.344 | 0.578 |
| KL | **0.660** | 0.358 | **0.551** | 0.751 | 0.127 | 0.749 | **0.902** | 0.084 | **0.414** | 0.609 | **0.343** | 0.579 |
| CrowdTruth | 0.633 | 0.643 | 0.547 | 0.756 | 0.122 | 0.750 | 0.897 | 0.315 | 0.336 | 0.603 | 0.354 | 0.589 |
| SLE (crowd) | 0.649 | 0.364 | 0.546 | **0.757** | **0.120** | **0.764** | 0.894 | 0.078 | 0.381 | 0.588 | 0.361 | **0.592** |

Table A.2. Hard and soft metric results on the real-world data sets. CIFAR-10S is the only real-world dataset that provides both annotator confidence and inter-annotator disagreement. ConvAbuse does not contain gold-standard labels. * MTLSL is trained using both gold and crowd annotations.

## A.7 Discussion

Results of the synthetic data experiments Figure A.1 show a clear benefit of using SLEs over majority and soft voting for label aggregation, especially when information regarding annotator confidence and reliability are available. Results on the real-world datasets Table A.2 are, however, mixed. SLEs with gold labels generally outperform standard cross-entropy training on gold labels as well as MTLSL, but this improvement may be an artifact of the different objective function. When using crowd annotations, CE and KL tend to outperform all other methods, although SLEs do perform best on the MRE dataset. As already discussed, a major limitation of these experiments is that none of the datasets contain all three sources of uncertainty. This means that even though SLEs are capable of modeling second-order uncertainty, they are not evaluated as such, since the targets contain only the types of uncertainty present in the datasets.

Future research should provide a more in-depth evaluation of the learned SLEs, such as their interpretability and ability to use post-hoc measures of reliability and annotator confidence. For example, we might compute a simple estimate of annotator reliability as each annotator's average percentage agreement with the others over all training examples, or a measure of confidence by computing the normalized entropy of the soft labels obtained from each annotator in CIFAR-10S.

# Appendix B

# Software

As part of the research completed for this thesis, we have developed two Python packages: PyBRAT and ExperimentConfig. This appendix briefly describes each package's purpose, benefits, and structure.

## B.1 PyBRAT

BRAT is a tool for annotating natural language text that has been used extensively for annotating NLP datasets due to its ease of use and ability to handle complex annotations [277]. Despite its ubiquity, however, existing software for working with BRAT annotations in Python –one of the most popular programming languages for ML and NLP research– is *ad hoc* and generally dataset specific, leading to a large amount of single-use code and a high risk for error. To address this gap, we developed PyBRAT, a Python package for loading, viewing, and manipulating BRAT annotations. PyBRAT is small, simple to use, and has no external dependencies. It also supports aligning BRAT annotations with their source text, allowing users to easily map between text and annotations.

The source code for PyBRAT is available at `https://github.com/jvasilakes/pybrat`.

### B.1.1 Structure

BRAT defines three types of annotations: spans, events, and attributes. The structure of PyBRAT mirrors this, defining classes for each annotation type which are themselves subclasses of a private `Annotation` class.

**Spans:** Spans are character spans in the text. They can be non-contiguous and can be associated with a label. An example span annotation in BRAT format is given below.

```
T1	Gene 58 63	CXCR1
```

where `T1` is the span ID, `Gene` is the label, `58 63` are the start and end character indices, and `CXCR1` is the text.

The corresponding PyBRAT `Span` class contains members for each of these fields, as well as an optional field specifying the source `.ann` file in which this span is found.

**Events:** Events in BRAT are collections of one or more Spans with an associated label and an optional function for each Span. For example,

```
E0      ASSOCIATED_WITH:T0 Subject:T1 Object:T2
```

where `E0` is the event ID, and the following `function:ID` pairs specify the members of this event and their function within the event. Because the structure of an Event is defined by the annotation task it can be arbitrarily complex. Events can also be nested, such that one Event contains another as a member. In the following example, the the Theme of the positive regulation event `E1` is another positive regulation event `E0`.

```
E0      Positive_regulation:T89 Theme:T3 Cause:T1
E1      Positive_regulation:T90 Theme:E0 Cause:T4
```

The PyBRAT `Event` class is therefore a simple collection of an arbitrary number of `Span` or `Event` instances, with additional members specifying the event ID, label, and source file.

**Attributes:** Spans and events can be assigned attributes, which describe additional information not covered by the span or event structure. For example,

```
A1      Polarity E1 Positive
```

where `A1` is the attribute ID, `Polarity` is the attribute type, `E1` is the associated annotation, and `Positive` is the attribute value. The `Attribute` class thus defines a single attribute type-value pair and links this to an existing `Annotation` instance.

#### BratAnnotation

The power of PyBRAT lies in its ability to read, write, and programmatically manipulate BRAT annotations as Python objects. This functionality is defined in the `BratAnnotations` class. `BratAnnotations` has a factory method `from_file()` that reads in a BRAT formatted `.ann` file, parses the spans, events, and attributes into their associated classes, and then links these class instances together according to the structure specified in the file. For example, given the BRAT file `example.ann`

```
T0      DiseaseOrSyndrome 51 72 myocardial infarction
T1      OrganicChemical 21 32   Clopidogrel
E0      TREATS OrganicChemical:T1 DiseaseOrSyndrome:T0
A0      Polarity E0 Positive
```

BratAnnotations.from_file("example.ann") would first create Span instances for T0 and T1. These would become the members of an Event instance for E0, and then an Attribute instance for A0 would be attached to E0 as an attribute. BratAnnotations has methods for iterating over spans, events, and attributes, retrieving annotations by ID, and filtering them by type (e.g., retrieving all of the TREATS events). It also has methods for adding or removing annotations and for rendering the annotations as a BRAT-formatted string and saving them to a file.

#### BratText

BRAT annotations necessarily have an associated text file which contains the annotated text. BratText defines methods for reading and writing this text, as well as functionality for cross-lookup between text and annotations. It includes basic functionality for tokenization and sentence splitting by default, but also allows complete customization of this functionality via user-defined functions, other Python packages, or external software.

Initialization of a BratText instance requires either a string of plain text, a JSON formatted list of sentences, or both. When providing a list of sentences, the required JSON format is

```
{"sent_index": int,
 "start_char": int,
 "end_char": int,
 "_text": str}
```

If both plain text and sentences are provided, the BratText initialization logic will ensure that the plain text can be reconstructed from the sentence list. Once created, the BratText instance defines methods for retrieving subspans of the text or individual sentences by Annotation instance (described in the Usage section below). This functionality is incredibly useful for constructing training instances for ML models. For example, it is possible to construct a dataset of Event-Sentence pairs by simply iterating over the Event instances in BratAnnotations and using BratText to retrieve the corresponding sentence, which can be passed to additional code that formats the pair as required by the model.

### B.1.2 Usage

Given the BRAT annotations file example.ann, it is possible to iterate over the annotations like so,

```
>>> from pybrat import BratAnnotations
>>> anns = BratAnnotations.from_file("example.ann")
>>> for event in anns.events:
>>>     ...  # process event
>>>     for event_attr in event.attributes:
>>>         ... # process event attribute
>>>     for span in event.spans:
>>>         ... # process span
>>>         for span_attr in span.attributes:
>>>             ... # process span attribute
```

By default, the Python `print` function will render the `Annotation` or `BratAnnotations` instance into a BRAT formatted string. For example,

```
>>> print(anns)
"""
T0      DiseaseOrSyndrome 51 72 myocardial infarction
T1      OrganicChemical 21 32   Clopidogrel
E0      TREATS OrganicChemical:T4 DiseaseOrSyndrome:T5
A0      Polarity E0 Positive
"""
>>> print(anns.events[0])
"E0      TREATS OrganicChemical:T4 DiseaseOrSyndrome:T5"
```

Manipulating individual `Annotation` instances is done by simply changing the value in question (e.g., `span.text = "new text"`). Manipulating a collection of annotations is done via the `BratAnnotations.add_annotation()` method, which takes a single `Annotation` instance as an argument. For example, we can add a new `Span` to `example.ann` like so,

```
>>> new_ann = Span(_id="T2", indices=[84, 92],
                   text="patient", _type="Human")
>>> anns.add_annotation(new_ann)
>>> print(anns)
"""
T0      DiseaseOrSyndrome 51 72 myocardial infarction
T1      OrganicChemical 21 32   Clopidogrel
T2      Human 84 92    patient
E0      TREATS OrganicChemical:T1 DiseaseOrSyndrome:T0
A0      Polarity E0 Positive
"""
```

Using `BratText` is similarly straightforward. Assuming we have the `example.ann` BRAT file as above as well as the text file `example.txt` and a JSON lines formatted list of sen-

tences in the text file `example_sentences.jsonl`, obtaining the text for a given annotation is as simple as passing that annotation instance to one of the `text()`, `tokens()`, or `sentences()` methods of `BratText`. For example,

```
>>> anntxt = BratText.from_file(text="example.txt",
                                sentences="example_sentences.jsonl")
>>> # The text between the start and end indices of the member spans.
>>> anntxt.text(anns.events[0])
"Clopidigrel successfully treated myocardial infarction"
>>> # The tokens of each member span.
>>> anntxt.tokens(anns.events[0])
["Clopidigrel", "treated", "myocardial", "infarction"]
>>> # The sentence which expressed the event.
>>> anntxt.sentences(anns.events[0])
"""
Clopidigrel successfully treated myocardial infarction in
patients receiving fibrinolytic therapy.
"""
```

## B.2 ExperimentConfig

Implementing and experimenting with ML models is often highly iterative. Starting with a simple baseline model, we continuously add features and run experiments or tests to validate functionality. Then once the final version of a model is implemented, experimentation requires running a number of experiments with different hyper-parameter values. Through this process it is crucial to log hyper-parameter values and code changes to accurately track and report results.

ExperimentConfig is a Python package meant to simplify the creation, manipulation, and logging of YAML-formatted config files for ML experiments. At its core, ExperimentConfig can be viewed as an extension to YAML in that it allows users to define custom validation logic and self-documentation for each hyper-parameter. In addition to the API, which is used within Python for reading and writing the config files, it also includes command-line functionality for generating template configs, validating configs, and updating parameter values. Finally, git integration is included by default, automatically linking each instance of a config file to a git commit hash, allowing users to easily match experiments to specific code versions, investigate discrepancies, and roll back changes.

The source code is available at `https://github.com/jvasilakes/experiment-config`.

### B.2.1 Structure

The only public class in ExperimentConfig is the `Config` class, which organizes parameters into groups and provides methods for saving, loading, and manipulating YAML config files. The workhorse of the `Config` class is the `Config.parameter` method, which is a function decorator that manipulates Python functions into `Parameter` instances. That is, the decorated function defines a single parameter alongside optional logic which restricts the values of the parameter, and these restrictions can be as simple or complex as the user is able to implement checks for. For example, one could write validation logic which insists that a parameter value is a nested array of integers between 2 and 42, nested exactly 3 arrays deep, with each member array containing no more than 10 items. Parameter validation is defined at the level of individual parameters via the code in the decorated function, but there is capability for high-level validation via the definition of post-load hooks. An example of this is given in the Usage section below.

Within a `Config` instance, `Parameter` instances are organized into groups as defined by the user, and these parameter groups can be nested. The `Parameter` class pairs a parameter name and a value, with additional logic around type and value restrictions, default values, deprecation status, and any comments.

Besides the API provided by the `Config` class, ExperimentConfig also defines command line functionality for working with config files on the command line via the `get_and_run_config_command()` function. This is explained in more detail in the following Usage section.

### B.2.2 Usage

Usage of ExperimentConfig can be broadly categorized into the creation of config files and the use of an existing config file within Python.

#### Creating Config Files

Using ExperimentConfig starts with creating a `.py` file that imports the public `Config` class and `get_and_run_config_command()` function, and then initializing a new `Config` instance.

```
from experiment_config import Config, get_and_run_config_command
config = Config("MyConfig")
```

We then define parameters for the `config` instance using the `@config.parameter` function decorator. The name of the decorated function will be the name of the parameter. The arguments to the decorator specify the parameter's group membership (subgroups are denoted using dot syntax, e.g., `Group.SubGroup`) as well as accepted data types and an an optional

default value. The function definition provides optional parameter value validation logic beyond its data type (e.g., value restrictions). Additionally, function docstrings are saved as documentation for the associated parameter and saved to the resulting YAML.

For example, let us assume that we are working with a simple encoder-decoder model. We want to name this experiment (e.g., to help with logging) and to define the structure of our encoder and decoder layers. Accordingly, we will have a generic `Experiment` parameter group which holds the name parameter as well as a `Model` group, which contains `Encoder` and `Decoder` subgroups for defining these layers, respectively.

```
@config.parameter(group="Experiment", types=str)
def name(val):
    assert len(val) > 0

@config.parameter(group="Model.Encoder", types=int, default=10)
def input_dim(val):
    assert val > 0

@config.parameter(group="Model.Encoder", types=int, default=5)
def output_dim(val):
    """
    This must equal Model.Decoder.input_dim
    """
    assert val > 0

@config.parameter(group="Model.Decoder", types=int, default=5)
def input_dim(val):
    assert val > 0

@config.parameter(group="Model.Decoder", types=int, default=3)
def output_dim(val):
    assert val > 0

@config.parameter(group="Model.Decoder", types=list,
                  default=[5, 3], deprecated=True)
def shape(val):
    assert len(val) == 2
    for member in val:
        assert isinstance(member, int)
        assert member > 0

@config.on_load
def validate_model_shapes():
    assert config.Model.Encoder.output_dim == config.Model.Decoder.input_dim
```

The `name` parameter belongs to the `Experiment` group and must be a non-empty string. The

following functions define the input and output dimensionalities of the encoder and decoder layers, which must be positive integers. Finally, the `@config.on_load` decorator defines a post-load hook that ensures the dimensionalities of the encoder and decoder layers are compatible. We also have defined a `shape` parameter that is marked as deprecated (presumably it used to specify the same information as `input_dim` and `output_dim`). Deprecated parameters function the same as non-deprecated ones, but raise a deprecation warning during validation.

At the end of the file, we use the following code to implement the command-line functionality.

```
if __name__ == "__main__":
    get_and_run_config_command(config)
```

The `get_and_run_config_command` function defines three commands: `new`, `validate`, and `update`. The `new` command generates a YAML config file with structure and default values according to the definitions in the `.py` file. So, running `python myconfig.py new myconfig.yaml` on the command line would generate the following YAML and save it to `myconfig.yaml`

```yaml
Experiment:
  name:
Model:
  Encoder:
    input_dim: 10
    # This must equal Model.Decoder.input_dim
    output_dim: 5
  Decoder:
    input_dim: 5
    output_dim: 3
    size: [5, 3]  # (deprecated)
Git:
  branch: main
  commit: '5169643'
  url: https://github.com/user/my_repo.git
```

Note that the `Git` parameter group is not defined in the `.py` file. The `Config` class automatically pulls this information from the current working directory and constructs the relevant parameters. If the current working directory is not part of a git repository, a warning is raised and this information is not included.

The `validate` command runs the validation code for each parameter as well as any post-load hooks defined by the `@config.on_load` decorator. Running `python myconfig.py validate myconfig.yaml` on the above file would raise an error, as the `Experiment.name`

parameter should not be empty. To update this parameter we can use either a text editor or the `update` command like so,

```
python myconfig.py update -f myconfig.yaml \
                          -p Experiment.name "experiment1"
```

where `-f` specifies a list of files to modify and `-p` specifies a single parameter-value pair to update. The `-p` flag can be used multiple times in a single call to update multiple parameter values as once. Coupled with a file glob as an argument to the `-f` flag (e.g., `-f config_files/*.yaml`) it is possible to update multiple parameters in a large number of config files with a single command.

Running the `update` command without any `-p` arguments simply updates the specified YAML files to match the config definition in `myconfig.py`. This is useful in multiple scenarios, such as when a new parameter has been added to the config definition but is not yet included in the YAML files, when a parameter becomes deprecated, or when the validation logic for an existing parameter changes.

#### Using Config Files

Once a config file has been constructed, it can be used within Python by importing the `config` instance (which defines the structure) and loading the YAML. For example, if we have our config definition in `myconfig.py` and valid values in `myconfig.yaml`, we can do the following,

```
>>> from myconfig import config
>>> config.load_yaml("myconfig.yaml")
```

`config` now contains all parameter values defined in the YAML, and these can be accessed through their parameter groups via the dot syntax.

```
>>> print(config.Model.Encoder.input_dim)
Parameter(input_dim, value=10)
>>> print(config.Model.Encoder.input_dim.value)
10
```

### B.2.3 Additional Thoughts

ExperimentConfig makes it easy to keep track of large numbers of parameters. Thus, when using ExperimentConfig one is encouraged to define parameters for as many variables as possible. Ideally, the config would contain all the information required to run a single experiment, meaning that one could simply pass a YAML file to a Python script without any

additional command-line arguments. After running the experiment, the Python script could then save the config file to a logging directory so that the experiment could be reproduced at a whim. Additionally, the inclusion of git commit information in the config file would allow reproducing experiments even from previous code versions by rolling back to the commit hash.

# Appendix C

# Evaluating LLMs for Classifying Polarity and Uncertainty

As discussed in Chapter 3, LLMs can be queried to provide additional views on SemMed Context for the purposes of generating large numbers of weak labels. We here discuss a preliminary evaluation of Mistral and Gemma 2 on the polarity and uncertainty tasks of the SemMed Context validation set.

## C.1 Methods

We evaluate two LLMs: Mixtral 8x7B Instruct[1] and Gemma 2 9B Instruct[2], both freely-available LLMs that achieve near SOTA performance on a variety of tasks. We programmatically query the models via the Huggingface Transformers library[3] using the following prompt template:

> Given the following sentence, determine if the relationship between the two entities in the sentence is negated/certain or not negated/uncertain. Provide the label only without any additional explanation. Answer in the following template: Answer: {}. Sentence: <sentence>, Entity 1: <entity>, Entity 2: <entity>

where the possible labels (negated vs. certain) depend on the task under evaluation.

Due to the large number of validation examples in SemMed Context and the high cost of inference of the LLM, we evaluate the model on two randomly selected subsets of 100 examples from the validation set –a subset for the polarity and uncertainty tasks, respectively– stratified to contain equal numbers of examples belonging to each label. While the prompt specifies a preferred answer format, we found that the LLMs often go "off-script" and output an explanation before the answer. We therefore obtain the predicted labels by a simple regular expression search for the first occurrence of a valid label in the output. In the few cases where the LLM does not predict a valid label, we take the most common label in the validation set.

[1] https://mistral.ai/news/mixtral-of-experts/
[2] https://blog.google/technology/developers/google-gemma-2/
[3] https://huggingface.co/

## C.2 Results and Discussion

| | Polarity | | | | | | Uncertainty | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Negative | | | Positive | | | Uncertain | | | Certain | | |
| | P | R | F | P | R | F | P | R | F | P | R | F |
| Mixtral | 0.909 | 0.400 | 0.555 | 0.615 | 0.960 | 0.750 | 0.581 | 0.640 | 0.610 | 0.600 | 0.540 | 0.568 |
| Gemma 2 | 0.796 | 0.780 | 0.788 | 0.784 | 0.800 | 0.792 | 0.512 | 0.840 | 0.636 | 0.556 | 0.200 | 0.294 |

Table C.1. Results of Mixtral 8x7B Instruct and Gemma 2 9B Instruct evaluated on subsets of 100 examples from the SemMed Context validation set, stratified to contain equal examples of each label for the Polarity and Uncertainty tasks, respectively. Predictions from each model were obtained zero-shot.

As the results in Table C.1 show, performance varies considerably across tasks and models. On the Polarity task, Mixtral obtains high precision on the Negative label and Gemma 2 even outperforms the best FactVAE model on this label (cf. Table 5.11, if we assume performance would generalize when evaluated on the full validation set). However, both models underperform on the Positive label, where our FactVAE obtained F1 scores $> 0.99$. On the Uncertainty task, both LLMs underperform our models on both labels. Gemma 2, especially, obtains only 0.294 F1 on the Certain label compared to F1s $> 0.98$ obtained by FactVAE.

Nevertheless, these models may still provide useful signals for reducing noise in SemMed Context within a programmatic weak supervision setup. It may also be possible to boost the performance of these LLMs using in-context few-shot learning, although an investigation of this was precluded by computation constraints.

Additionally, while Mixtral and Gemma 2 are modern, high performing LLMs, they are far from the most powerful available. Again, computational constraints barred us from additional experiments, but we suspect that very large models such as Llama 3 (70B parameters), Mistral Large (123B parameters), and ChatGPT 4 (1.8T parameters) would perform better on these tasks.