Neural Token Representations and Negation and Speculation Scope Detection in Biomedical and General Domain Text

Elena Sergeeva
MIT EECS
elenaser@mit.edu
Amir Tahmasebi
CodaMetrix
amir.tahmasebi@gmail.com

Henghui Zhu
Boston University
henghuiz@bu.edu
Peter Szolovits
MIT EECS
psz@mit.edu

Abstract

Since the introduction of context-aware token representation techniques such as Embeddings from Language Models (ELMo) and Bidirectional Encoder Representations from Transformers (BERT), there have been numerous reports on improved performance on a variety of natural language tasks. Nevertheless, the degree to which the resulting context-aware representations can encode information about morpho-syntactic properties of the tokens in a sentence remains unclear.

In this paper, we investigate the application and impact of state-of-the-art neural token representations for automatic cue-conditional speculation and negation scope detection coupled with the independently computed morpho-syntactic information. Through this work, We establish a new state-of-the-art for the BioScope and NegPar corpora.

Furthermore, we provide a thorough analysis of neural representations and additional features interactions, cue-representation for conditioning, discussing model behavior on different datasets and, finally, address the annotation-induced biases in the learned representations.

1 Introduction

In 2018, a set of new state-of-the-art results were established for a variety of Natural Language Processing tasks, the majority of which can be attributed to the introduction of context aware token representations, learned from large amounts of data with Language-modeling like tasks as a training goal (Devlin et al., 2018; Peters et al., 2018). It is, however, unclear to what degree the computed representations capture and encode high-level morphological/syntactic knowledge about the usage of a given token in a sentence. One way of exploring the potential of the learned representation would be through investigating the performance on a task that would require the representation to acquire some notion of syntactic units such as phrases and clauses, as well as the relationship between the syntactic units and other tokens in the model. An example of such a task is *Speculation* or *Negation Scope Detection*.

The main contributions of this work can be summarized as follows:

- We achieve and report a new state-of-the-art for the negation and speculation scope detection on several biomedical and general domain datasets, which were created using different definitions of what constitutes a scope of a given negation/speculation.[1]
- We investigate different ways of incorporating additional automatically-generated syntactic features into the model and explore the potential improvements resulting from the addition of such features.
- Following Fancellu et al. (2017), we provide a thorough comparison of our proposed model with other state-of-the-art models and analyze their behaviour in the absence of potential "linear clues", the presence of which might result in highly accurate predictions even for syntax-unaware token representations.

2 The Task

In general, speculation or negation scope detection can be constructed as the following conditional token classification task: given a negation or speculation **cue** (*i.e.*, a word or phrase that expresses negation or speculation such as 'No' and 'May'),

[1] An implementation of our model together with the pre-trained models for scope detection will be available later.

Proceedings of the 10th International Workshop on Health Text Mining and Information Analysis (LOUHI 2019), pages 178–187
Hong Kong, November 3, 2019.
https://doi.org/10.18653/v1/D19-62

identify which tokens are affected by the negation or represent an event that is speculative in nature (referred to as **the scope of the negation or speculation**). Consider the following example:

> (1) These findings that (***may*** *be from an acute pneumonia)* include minimal bronchiectasis as well.

In this case, the speculation cue is **"may"** and the string of tokens that contains the speculative information is **"may be from an acute pneumonia"**.

Each data point, as such, is a string of tokens paired with the corresponding negation or speculation cue. Note that nested negations in the same sentence would be distinguished only by the associated cue.

From the syntactic structure point of view, it is clear that in most cases, the boundaries of a given scope strongly correlate with the clausal structure of the sentence (Morante and Sporleder, 2012) There is also a strong connection between the fine-grained part-of-speech (POS) of the cue and the scope boundaries.

Consider the following examples where the type of possible adjectives (either attributive or predicative) results in different scope boundaries (scope highlighted as italic):

> (2) This is a patient who had ***possible*** *pyelonephritis* with elevated fever.
>
> (3) *Atelectasis in the right mid zone is, however,* ***possible***.

Such a property of the task requires a well-performing model to be able to determine cue-types and the corresponding syntactic scope structure from a learned representation of cue-sentence pairs. As such, it can be used as an (albeit imperfect) proxy for assessing the knowledge about the structure of the syntax that a sentence aware token representation potentially learns during training.

2.1 Datasets

There are no universal guidelines on what constitutes a scope of a given negation or speculation; different definitions might affect a given model's performance. To take this ambiguity into account, we report our results on two different datasets: BioScope (Vincze et al., 2008) and NegPar (Liu et al., 2018).

- The BioScope corpus (Vincze et al., 2008) consists of three different types of text: Biological publication abstracts from Genia Corpus (1,273 abstracts), Radiology reports from Cincinnati Children's Hospital Medical Center (1,954 reports), and full scientific articles in the bioinformatics domain (nine articles in total). In this work, we focus on two of the sub-corpora: Abstracts and Clinical reports. One should note that BioScope corpus does not allow discontinuous scopes.

- NegPar (Liu et al., 2018) is a corpus of Conan Doyle stories annotated with negation cues and the corresponding scopes. The corpus is available both in English and Chinese. In this work, we only use the English part of the corpus. Unlike BioScope, NegPar provides a canonical split as training (981 negation instances), development (174 instances) and test sets (263 negation instances). NegPar annotation guidelines allows for discontinuous scopes.

3 Previous Work

Negation scope detection algorithms can be classified into two categories: (1) rule-based approaches that rely on pre-defined rules and grammar; and (2) statistical machine learning approaches that utilize surface level features of the input strings to detect the scope of the negation.

Rule-based approaches Due to the somewhat restricted nature of clinical texts syntax, a pre-defined rule-based key-word triggered negation scope detection system achieves competitive performance on a variety of clinical-notes derived data-sets (Chapman et al., 2001; Harkema et al., 2009; Elkin et al., 2005).

Machine learning approaches While rule-based approaches might achieve high performance on medical institution specific datasets, they do not generalize well for other dataset types and they may require customization of the rules to adapt to the new corpus and/or domain. By contrast, machine learning-based systems do not require active human expert participation to adapt to a new dataset/domain. Earlier works utilizing the statistical approaches for negation scope detection include Support Vector Machines (SVM), Conditional Random Fields based models (CRF) (Agarwal and Yu, 2010; Councill et al., 2010) as well

as hybrid CRF-SVM ensemble models (Zhu et al., 2010) (Morante and Daelemans, 2009)

Recently, Neural Network-based approaches have been proposed for such tasks, including Convolutional Neural Network (CNN)-based (Qian et al., 2016) and Long Short Term Memory (LSTM)-based (Fancellu et al., 2017; Sergeeva et al., 2019) models.

The work on specifically speculation scope detection is less varied and mainly confined to CONLL-2010 Shared-Task2 submissions (Farkas et al., 2010). It is, however, important to note that due to the similarity in the formulation of the task, the majority of the negation-specific machine learning approaches can be directly applied to the speculation scope detection problem provided the speculation annotated data is available for training.

We also draw inspiration from a large body of work (Linzen et al., 2016; Gulordava et al., 2018; Marvin and Linzen, 2018) examining the nature of modern context aware representations from a linguistic perspective.

4 Model Training and Evaluation

4.1 Neural Token Representation

The use of pre-trained continuous word representations has been ubiquitous in modern statistical natural language processing. The importance of an appropriate word-level representation is especially noticeable in per-token prediction tasks: in such a set-up the model goal is to fine-tune or modify the existing input token representation in such a way that it contains the necessary information to make a correct classification decision at prediction time.

In this work, we consider the following approaches for generating the input token representation:

- **Global Vectors (GloVe)** (Pennington et al., 2014): A pre-trained token representation that relies on the direct matching of tokens and the corresponding ratios of token co-occurrences with their neighbours. Note that the definition of the neighbour in this setup is static (that is, the ultimate representation would incorporate an averaged notion of context) and relies on the bag-of-words representation of the context.

- **Embeddings from Language Models (ELMo)** (Peters et al., 2018): A bi-directional LSTM model-based token representation, pre-trained on the language modeling task. Instead of modeling the bag-of-words neighborhood co-occurrence probabilities directly, this model approximates the conditional probability of a token given the ordered linear context of the token usage.

- **Bidirectional Encoder Representations from Transformers (BERT)** (Devlin et al., 2018): A transformer-based token representation trained on the modified language modeling task together with a broader context next sentence prediction task. In this model, the context of a token is continuously incorporated into the representation of the token itself as a weighted sum of the neighboring token representations through the use of the multi-head-attention mechanism. The linear order of the token information is provided at input time as an additional positional embedding, since the unmodified transformer architecture does not encode any notion of the linear order.

Despite the performance gains achieved by the widespread use of contextual word embeddings like ELMo and BERT, the questions about the nature of the learned representation remain unanswered. Both ELMo and BERT were introduced to incorporate the wider structure of the given input into individual token representation at the time of training; however, both models only have access to the linear order of the context.

The question then arises: To what degree does the word embedding trained on a language modeling like task and computed using the whole linear context of a sentence encode the broader syntax-related characteristics of a token used within a context?

In order to gain insight into the nature of the learned representations and their potential use for negation and speculation scope detection, we introduce the following syntax-informed features to be used together with the token embedding:

POS : Part-Of-Speech of a given token as defined by the Penn Treebank tagging scheme (Marcus et al., 1993).

DEP : Type of dependency between the token and its parent, representing limited dependency tree information of a given token.

PATH : A string of Google Universal POS tags (Petrov et al., 2012) of the three direct ancestors of the token in the dependency tree; this feature captures local constituent-like information of a given token.

LPATH : Depth of the token in the syntactic dependency tree.

CP : The distance between a given token and the negation cue in the constituency parse tree generated using (Kitaev and Klein, 2018). If a negation cue has multiple tokens, the minimum of the distances is used.

Note that all features were automatically generated, and as a result, represent a "noisy" source of information about the syntactic characteristic of a token. If adding syntactic features as additional inputs would not affect or would significantly degrade the model's performance, it is reasonable to assume that the information represented by such features is already present in the token representation in some way.

4.2 Modes of Evaluation

To provide a fair comparison of different types of embeddings, we introduce two different modes of evaluation. The first mode (referred to as **Feature-based embeddings** later in the paper) is designed to test the embeddings in the same setup as previously used to get the state-of-the-art performance on the dataset. The second mode (referred to as **BERT fine-tuning** later in the paper) is designed to test BERT embeddings in their native direct fine-tuning setting.

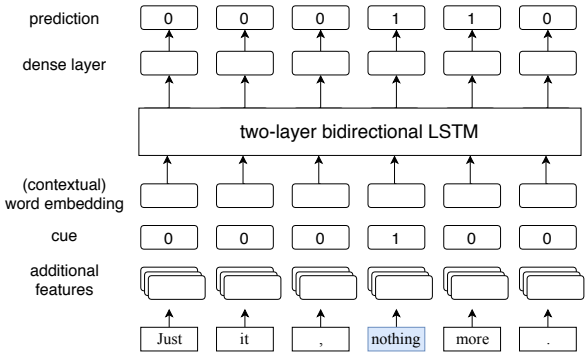

Figure 1: A diagram of the proposed bi-directional LSTM model for negation and speculation detection with additional features.

Feature-based Embeddings using Bi-directional LSTM: Figure 1 demonstrates the proposed framework for the desired task. One should note that the factor that differentiates the two experiments from one another is the embeddings. The task specific layers (two-layer Bi-directional LSTM) remains the same across all experiments. To properly condition each scope on a given cue, we concatenate a specific cue embedding to the input embedding, before computing the final representation for each token. Additional syntactic information is also provided by concatenating the input embedding with all of the syntactic feature embeddings.

BERT Fine-tuning: The original setup for the use of BERT embedding does not require an elaborate task-specific layer; the task specific model is a copy of the original transformer-based BERT architecture with the corresponding pre-trained model-parameters, and the top prediction layer swapped for a new task specific layer that predicts the probability of a given label for a token representation. Crucially, the token representation is allowed to change during the fine-tuning. For this particular setup, it is unclear how to account for the conditional nature of the scope prediction task. In other words, a sentence can potentially contain more than one negation/piece of speculative information.

We consider two different testing scenarios to evaluate the different ways of providing the cue information to the model:

1. Providing the embedded cue at the top layer of the model by concatenating it to the learned token embedding.

2. Providing the embedded cue at the bottom as a part of the input to the transformer layer before the fine-tuning by adding the cue embeddings (initialized randomly at the fine-tuning stage) to the initial token representation.

To test if the additional syntactic information provides any additional benefit to our framework, we also add the mean of all of the syntactic feature embeddings to the initial pre-transformer representation of the input.

4.3 Hyperparameter Settings

Feature-based Embeddings For the aforementioned set of experiments, the following architecture parameters have been considered:

Table 1: Performance of the negation scope detection task on BioScope and NegPar corpora using different approaches. Results are reported as the percentage of number of predicted scopes that exactly match the golden scope (PCS)

Model	BioS_Abstracts	BioS_Clinical	NegPar	NegPar(CV)
Fancellu et al. (2017)	81.38%	94.21%	68.93 % [a]	N/A
Fancellu et al. (2018)	N/A	N/A	61.98% [b]	N/A
Bi-LSTM$_{\text{GloVe}}$	63.24%(1.80%)	90.46%(3.64%)	51.48%(4.45%)	49.18%(4.97%)
Bi-LSTM$_{\text{ELMo}}$	81.62%(1.87%)	93.10%(2.18%)	71.52%(1.98%)	75.29%(3.35%)
Bi-LSTM$_{\text{BERT}}$	79.29%(3.06%)	91.26%(2.82%)	66.78%(3.50%)	69.45%(3.55%)
Bi-LSTM$_{\text{GloVe}}$ + AF	79.00%(2.07%)	94.02%(1.98%)	69.70%(2.81%)	73.11%(3.19%)
Bi-LSTM$_{\text{ELMo}}$ + AF	83.30%(3.16%)	**94.25%**(2.86%)	69.96%(2.12%)	75.43%(4.82%)
Bi-LSTM$_{\text{BERT}}$ + AF	80.68%(3.23%)	93.10%(2.77%)	67.42%(2.10%)	73.39%(4.12%)
BERT (c-top)	74.63%(3.23%)	92.87%(2.04%)	63.14% (2.08%)	—
BERT (c-bottom)	86.97%(2.24%)	93.68%(2.37%)	76.78%(2.04%)	81.91%(3.04%)
BERT (c-bottom) + AF	**87.03%**(2.38%)	93.45%(1.63%)	**79.00%**(1.37%)	80.64%(2.57%)

The number in the parenthesis indicates the standard deviation of the score.

[a] These results are generated using an older version of the corpus annotation.

[b] Since this work is aimed at cross-lingual negation detection, the reported results are based on using cross-language word embeddings, which are likely to degrade a single-language model performance.

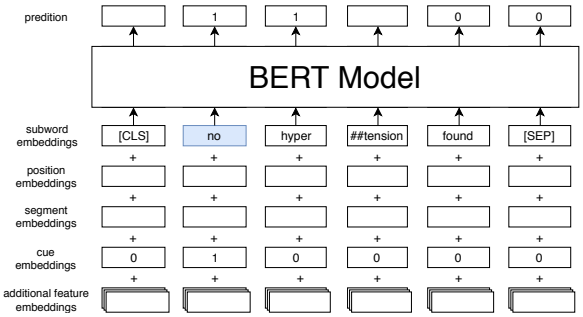

Figure 2: A diagram of the proposed BERT-based architecture for negation and speculation scope detection with inclusion of additional features.

- Word embedding dimension: GloVe: 300; ELMo, BERT: 1024
- Syntactic feature embedding dimension: 10 per feature
- Task-specific LSTM embedding dimension: 400

During training, a dropout rate of 0.5 (Gal and Ghahramani, 2016) was used to prevent overfitting. The Adam optimizer (Kingma and Ba, 2014) was used with step size of 10^{-3} and batch size of 32 for 50 epochs for BioScope and 200 epochs for NegPar. The reason why we use different epochs is that there are fewer training examples for NegPar than BioScope. Therefore, it takes more epochs for the NegPar models to converge.

BERT Fine-tuning The BERT models have the following architecture parameters:

- Word embedding dimension: 1024
- BERT$_{\text{LARGE}}$ layer transformer configuration (Devlin et al., 2018)
- Syntactic features embedding dimension: 1024 for each feature
- Cue embedding dimensions: 1024

We perform fine tuning on the negation/speculation task for 20 epochs. The Adam optimizer was used with learning rate of 10^{-5} and batch size of 2 for 10 epochs for the BioScope corpus and 50 epochs for the NegPar corpus.

4.4 Evaluation Procedure

We report our results in terms of the percentage of number of predicted scopes that exactly match the golden scopes (PCS). Since pre-trained BERT models use their own tokenization algorithm, it results in inconsistent final number of tokens in the dataset across evaluation modes. As a result, other traditional evaluation metrics such as precision, recall and F1 are inappropriate to be used in this study as they depend on the number of tokens.

Since the BioScope dataset does not have a canonical training/development/test set split, we report 10-fold cross-validation results together with the standard deviation of the resulting scores.

For the NegPar dataset, we report the result on the test set as well as 10-fold cross-validation results. To overcome the possible random initialization influences on the results, we report the average score for 10 random seeds on the test set together with the associated standard deviation.

5 Results

The performance of different approaches on BioScope and NegPar corpora for the negation scope detection and the speculating scope detection are shown in Table 1 and Table 2, respectively. Bi-LSTM-marked entries of the table correspond to Feature-based and BERT-marked entries correspond to BERT fine-tuning approaches.

5.1 Feature-based Approach

Effect of embedding on performance: Except for the negation scope detection task on BioScope clinical notes, ELMo embeddings significantly outperformed GloVe embeddings as well as the feature-based use of BERT embeddings, but not the fined-tuned version of BERT. While the former is expected, the latter is noteworthy: for NER task (Devlin et al., 2018), for example, the difference in performance between the fine-tuning and feature-based approach results is 1.5% of the F1 score. For negation scope detection the difference is a striking 7.68% on BioScope-abstracts and 10% on a test set of the NegPar dataset. For speculation scope detection the difference remains as large (7.93%). We theorize that this differences comes from the different syntactic nature of the target strings of tokens: NER systems are concerned with finding named entities in text, where the majority of the named entities are represented by relatively short (token-wise) noun phrases, negation/speculation scope detection requires recognition of a much more diverse set of syntactic phenomena. This suggests an important difference between the featurized and fine-tuned approaches for highly syntax-dependent token classification tasks.

Syntactic features induced gains: In general, we observe consistent small gains in performance for all types of embedding on BioScope (both speculation and negation detection modalities) but not on the NegPar dataset. The only exception to this pattern is in non-context aware GloVe embeddings. Adding syntactic features embeddings has inconsistent effects on standard deviations over modalities and datasets.

5.2 BERT fine-tuning approach

Cue-conditioning influence on the results The way to condition a given instance on a particular cue greatly influences the model performance: providing cue information at the top layer of the model results in poor performance of the model for all datasets and both negation and speculation modalities.

Syntactic features induced gains and the importance of Cross Validation evaluation: Adding features to the best performing BERT fine-tuned models does not result in any significant differences on the BioScope dataset. We observe a significant gain in performance on NegPar: note that in this case the gain is purely train/test set split induced and disappears entirely in a cross-validation mode of evaluation.

Artificial noise and the model performance: Even though the experimental results suggest no to minimal contribution of the additional features to the best model performance, natural questions to ask are: "Does the feature enriched model rely on the provided features during the prediction phase?" and "Do the final learned representations differs significantly for feature-enriched and featureless inputs?" We introduce noise into the trained model inputs to check if artificial noise undermines its performance. In particular, we consider the model BERT(cue-bottom) + AF, as it provides the best performance out of all feature-enriched models.

With a given probability, which we call the noise level, we replace a given feature value with a random value: for categorical features (POS, DEP, PATH), we replace it with a random category, and for numerical features (LPATH,CP), we replace it with a random integer drawn from a uniform distribution bounded by the feature's possible minimum and maximum values. We observe a consistent and significant decrease in performance as the probability of seeing the incorrect features increases (see Figure 3). This suggests that the additional features introduced in this paper play an important role in decision making. This is supported by the fact that the performance on clini-

Table 2: Performance of speculation scope detection task on BioScope corpus using different approaches. Results are reported as the percentage of number of the predicted scopes that exactly match the golden scope (PCS).

Model	BioScope_Abstracts	BioScope_Clinical
Qian et al. (2016)	85.75%	73.92%
Bi-LSTM$_{\text{GloVe}}$	47.99%(4.07%)	46.90%(2.87%)
Bi-LSTM$_{\text{ELMo}}$	84.62%(2.33%)	81.82%(2.74%)
Bi-LSTM$_{\text{BERT}}$	81.35%(1.95%)	78.75%(3.24%)
Bi-LSTM$_{\text{GloVe}}$ + AF	85.07%(2.66%)	80.73%(3.01%)
Bi-LSTM$_{\text{ELMo}}$ + AF	86.57%(2.65%)	81.55%(2.74%)
Bi-LSTM$_{\text{BERT}}$ + AF	84.43%(1.08%)	81.37%(4.32%)
BERT (cue-top)	57.32%(2.14%)	60.49%(4.77%)
BERT (cue-bottom)	**89.28%(1.65%)**	**83.71%(2.77%)**
BERT (cue-bottom) + AF	88.91%(1.65%)	82.36%(4.27%)

The number within the parentheses indicates the standard deviation of the score.

cal reports negation detection remains almost unaffected by the change, since the majority of the negation scopes in this dataset can be captured by structure-independent heuristics.

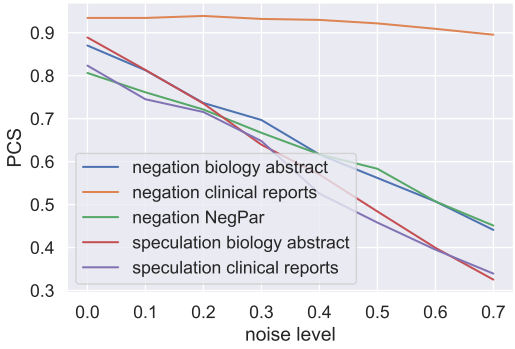

Figure 3: Performance of the BERT + AF models with respect to the noise level, averaged for 10 fold CV

5.3 Linear punctuation cues and model performance

Even though the scope boundaries correlate to the syntactic structures of the sentence, a good performance on a given dataset does not necessarily prove the model acquired any kind of a structural knowledge: as was noted in Fancellu et al. (2017), the majority of scopes in the BioScope corpus consist of cases where the punctuation boundaries match the scope boundaries directly. For those cases, the model does not have to learn any kinds of underlying syntactic phenomena: learning a simple heuristic to mark everything between a cue and the next punctuation mark as a scope would produce an illusion of a more complex syntax-informed performance.

To see if our model's performance is significantly affected by the punctuation clues, we remove all the punctuation from the training corpus, re-train all the models on the modified dataset and evaluate the learned models on the test set. We also report the performance on "hard" (non-punctuation bound) instances of scopes separately.

As can be seen in Table 3, removing punctuation affects all models' behaviour similarly: model performance degrades by losing 2-3 percent of PCS on average. Interestingly, the performance on the non-punctuation boundaries scopes declines similarly, which suggest that punctuation plays an important role in computing a given token representation, and not only as a direct linear cue that signifies the scope's start and end.

5.4 Error overlap

Given the difference in the model architectures, a natural question to ask is: "Is the best performing model strictly better than the others, or do they make different types of errors?" We compute the error overlap between BERT and ELMo on the negation detection task as shown in Figure 4. About half of ELMo and slightly more than a quarter of BERT errors appear to be model specific, suggesting the potential for ensemble-induced improvements.

We also compute the error overlap for the NegPar test set performance for the top 3 performing models: almost half of the ELMo errors and about 3/4 of BERT fine-tuned and BERT fine-tuned with

Table 3: Performance on percentage of correct span on BioScope Abstracts sub-corpus trained under different schemes.

	Trained w/ punctuation		Trained w/o punctuation	
	all	hard cases	all	hard cases
Bi-LSTM$_{\text{GloVe}}$	63.24%(1.80%)	51.83%(3.47%)	57.19%(2.48%)	48.82%(3.36%)
Bi-LSTM$_{\text{ELMo}}$	81.62%(1.87%)	73.07%(3.60%)	76.90%(2.72%)	69.88%(3.78%)
Bi-LSTM$_{\text{BERT}}$	79.29%(3.06%)	70.37%(6.60%)	77.54%(3.18%)	70.28%(5.41%)
Bi-LSTM$_{\text{GloVe}}$ + AF	79.00%(2.07%)	68.79%(4.06%)	76.21%(1.76%)	67.96%(2.64%)
Bi-LSTM$_{\text{ELMo}}$ + AF	83.30%(3.16%)	75.56%(4.46%)	82.31%(2.95%)	76.58%(3.67%)
Bi-LSTM$_{\text{BERT}}$ + AF	80.68%(3.23%)	72.68%(6.71%)	80.45%(3.24%)	73.19%(5.52%)
BERT (c-bottom)	86.97%(2.24%)	**82.51%(3.78%)**	83.48%(3.22%)	**79.42%(4.46%)**
BERT (c-bottom +AF)	**87.03%(2.38%)**	82.38%(4.48%)	**84.58%(3.58%)**	79.27%(5.82%)

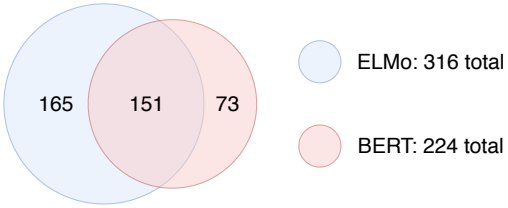

Figure 4: Distribution of error overlaps: BERT vs. ELMo on BioScope Abstracts dataset.

features are common for all of the models. It is interesting to note that the the errors of BERT without the features are not a subset of BERT with the features, suggesting the possibility of a performance trade-off instead of a straight feature-induced performance improvement.

Qualitatively, on average ELMo tends to prefer longer scopes, sometimes extending the scope for an additional clause. Both models have trouble with common words that can be encountered in a variety of different contexts, such as certain prepositions and personal pronouns.

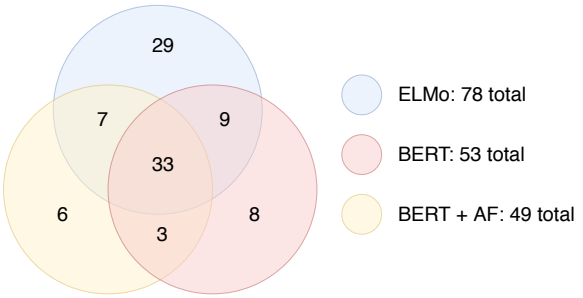

Figure 5: Distribution of error overlaps: BERT vs. BERT with features (BERT + AF) vs. ELMo on NegPar test set.

6 Conclusions and Future Work

This work presents a comparison among different context-aware neural token representations and the corresponding performance on the negation and speculation scope detection tasks. Furthermore, we introduce a new state-of-the-art BERT-based cue-conditioned feature-enriched framework for negation/speculation scope detection. Based on the empirical results, we are inclined to recommend BERT fine-tuning over using a feature-based approach with BERT for syntax-dependent tasks.

We used two commonly used publicly available datasets, BioScope and NegPar for our evaluation. Despite the observed gains on the test set of the NegPar corpus, the effect of the syntactic features on BERT (fine-tuned) performance remains largely inconclusive.

It is also important to note that the syntactic information we have been trying to incorporate into the model was generated automatically; one of the possible avenues of research would be comparing the possible golden annotation induced gains with the imperfect information gain we observe when incorporating silver syntactic features.

We were unable to find any consistent grammatical explanation for the errors context-aware models result in on the test data; however, this does not conclusively mean that such an explanation does not exist. An appropriate next step would be annotating a smaller set of sentences, grouped by the corresponding syntactic construction and see if a given token representation yields improved performance on such a construction.

Acknowledgments

Research partially supported by the Office of Naval Research under MURI grant N00014-16-1-2832

References

Shashank Agarwal and Hong Yu. 2010. Biomedical negation scope detection with conditional random fields. *Journal of the American Medical Informatics Association*.

Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34(5):301–310.

Isaac Councill, Ryan McDonald, and Leonid Velikovich. 2010. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 51–59. University of Antwerp.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Peter L. Elkin, Steven H. Brown, Brent A. Bauer, Casey S. Husser, William Carruth, Larry R. Bergstrom, and Dietlind L. Wahner-Roedler. 2005. A controlled trial of automated classification of negation from clinical notes. *BMC Medical Informatics and Decision Making*, 5(1):13.

Federico Fancellu, Adam Lopez, Bonnie Webber, and Hangfeng He. 2017. Detecting negation scope is easy, except when it isn't. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 58–63, Stroudsburg, PA, USA. Association for Computational Linguistics.

Federico Fancellu, Adam Lopez, and Bonnie L. Webber. 2018. Neural networks for cross-lingual negation scope detection. *CoRR*, abs/1810.02156.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Tex. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning – Shared Task*, pages 1–12. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29*, pages 1019–1027. Curran Associates, Inc.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. 2009. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839–851.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Qianchu Liu, Federico Fancellu, and Bonnie Webber. 2018. NegPar: a parallel corpus annotated for negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202. Association for Computational Linguistics.

Roser Morante and Walter Daelemans. 2009. A metalearning approach to processing the scope of negation. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 21–29. Association for Computational Linguistics.

Roser Morante and Caroline Sporleder. 2012. Modality and negation: An introduction to the special issue. *Computational linguistics*, 38(2):223–260.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Zhong Qian, Peifeng Li, Qiaoming Zhu, Guodong Zhou, Zhunchen Luo, and Wei Luo. 2016. Speculation and Negation Scope Detection via Convolutional Neural Networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 815–825, Stroudsburg, PA, USA. Association for Computational Linguistics.

Elena Sergeeva, Henghui Zhu, Peter Prinsen, and Tahmasebi Amir. 2019. Negation scope detection in clinical notes and scientific abstracts: A feature-enriched lstm-based approach. *AMIA Jt Summits Transl Sci Proc. 2019*, pages 212–221.

Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(S11):S9.

Qiaoming Zhu, Junhui Li, Hongling Wang, and Guodong Zhou. 2010. A unified framework for scope learning via simplified shallow semantic parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 714–724. Association for Computational Linguistics.