Negation Scope Refinement via Boundary Shift Loss

Yin Wu and **Aixin Sun**
Nanyang Technological University, Singapore
wuyi0023@e.ntu.edu.sg, axsun@ntu.edu.sg

Abstract

Negation in language may affect many NLP applications, *e.g.,* information extraction and sentiment analysis. The key sub-task of negation detection is *negation scope resolution* which aims to extract the portion of a sentence that is being negated by a negation cue (*e.g.,* keyword "not" and "never") in the sentence. Due to the long spans, existing methods tend to make wrong predictions around the scope boundaries. In this paper, we propose a simple yet effective model named **R-BSL** which engages the *Boundary Shift Loss* to refine the predicted boundary.[1] On multiple benchmark datasets, we show that the extremely simple R-BSL achieves best results.

1 Introduction

Negation is a complex linguistic phenomenon. Even though there does not exist a widely agreed task definition for negation detection, two sub-tasks are commonly performed: (i) *negation cue detection*, and (ii) *negation scope resolution*. Negation cue is a keyword (*e.g.,* not, never) in a sentence that acts as an indicator of semantic negation, and its detection is relatively easy. Negation scope refers to the portion(s) in a sentence being semantically affected (*i.e.,* negated) by the cue. There could be multiple cues in one sentence and each corresponds to its own scope. Table 1 lists three cues in the same sentence and their scopes.

Different datasets may adopt different annotation guideline of scopes, *e.g.,* whether or not a cue itself is a part of its scope. The example sentence in Table 1 well demonstrates the unique characteristics of this task compared to other span extraction tasks like Named Entity Recognition (NER). They are: (i) a negation scope is defined by (or associated to) a given cue, (ii) the negation spans are usually longer than a named entity, and (iii) a good number of negation spans are discontinuous, depending on the adopted annotation guideline.

In recent years, pretrained language models (PLMs) like BERT (Devlin et al., 2019) have been explored to improve negation detection (Khandelwal and Sawant, 2020; Khandelwal and Britto, 2020). Specially designed pre-training material that focuses on negation has also been explored and achieves state-of-the-art performance (Truong et al., 2022). Nevertheless, we believe that negation detection shall be considered as a pre-processing step for downstream subtasks and its model shall not be over-complicated.

In this paper, we enhance a simple baseline by Khandelwal and Sawant (2020) with an effective Boundary Shift Loss (BSL), to refine the predicted negation scope boundaries. BSL is derived based on the positions of span boundaries. For each token, boundary shift tells the direction of the nearest span boundary: left or right. With the simple BERT + Feed-forward architecture, our R-BSL model outperform baselines on all well-known datasets.

2 Related Work

Negation detection was firstly studied in biomedical and health texts, represented by NegEx (Chapman et al., 2001) developed for EHRs. NegEx is built on top of regular expressions; its negation scopes are mainly named entities. The definition of negation scope becomes largely different and more generic in later datasets. The BioScope corpus (Vincze et al., 2008) annotates negation scope in biological full papers and scientific abstracts. The "Sherlock" corpus (Morante and Blanco, 2012), annotates Conan Doyle's novels *Sherlock Holmes* series. SFU Review Negation corpus (Konstantinova et al., 2012) annotates negations and speculations in the SFU Review corpus (Taboada et al., 2006) for sentiment analysis.

Like many other NLP tasks, BERT leads to significant improvement on scope resolution (Khan-

[1] Our code is available at https://github.com/LuciusLan/BSL_Negation

Findings of the Association for Computational Linguistics: ACL 2023, pages 6090–6099
July 9-14, 2023

Cue	Negation scope marked in discontinuous "*span*" s
in-	Mr. Sherlock Holmes, who was usually very late in the mornings, save upon "*those*" not [cue] in- [/cue] "*frequent occasions when he was up all night*", was seated at the breakfast table.
not	Mr. Sherlock Holmes, who was usually very late in the mornings, save upon "*those*" [cue] not [/cue] "*infrequent occasions when he was up all night*", was seated at the breakfast table.
save	Mr. Sherlock Holmes, "*who was*" usually "*very late in the mornings*", [cue] save [/cue] "*upon those not infrequent occasions when he was up all night*", was seated at the breakfast table.

Table 1: An example sentence with three different negation cues, and their corresponding scopes. The cues are marked with special tokens [cue] cue [/cue], and scopes "*span*" s are in italic with double quotation marks.

delwal and Sawant, 2020). Results are further improved in later research (Khandelwal and Britto, 2020) with more advanced PLMs like RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019), together with multi-task training. Recently, Truong et al. (2022) utilize additional pre-training with negation cue masking, achieving better performance on BioScope and SFU, but poorer results on Sherlock. Nevertheless, the higher performance comes with the price of extra training resources and time.

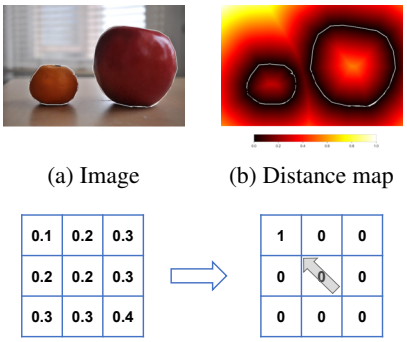
3 Problem Definition

As a common practice, we assume that negation cue has been successfully detected. Our key focus is *negation scope resolution* for a given cue. For presentation simplicity, we assume there is only one cue in a given sentence. The cases of multiple cues can be easily achieved by sentence duplication, each time with a different known cue being wrapped with special indicator tokens. The model would be trained to predict negation scope of each cue separately. Table 1 gives a typical example of how sentence with three negation cues and three corresponding scopes is being pre-processed by duplication and the special indicator tokens [cue] [/cue].

Given an input sequence $S = \langle t_1, t_2, ..., t_n \rangle$, with a known cue, the task is to predict the cue's negation score in token spans. We adopt the OSC tagging scheme: $Y = \langle y_1, y_2, ..., y_n \rangle$ where y_i is O if t_i is non-scope, S if t_i is part of the scope, and C if t_i is the given cue. We use a dedicated label "C" for cue, to satisfy the annotation guidelines in different datasets, *i.e.,* not all annotations consider cue as a part of the scope.

4 The R-BSL Model

The central idea of Boundary Shift Loss is inspired by techniques used for semantic segmentation.



(a) Image (b) Distance map

(c) local distance → direction map

Figure 1: (a) A sample image and (b) its normalized boundary distance map. (c) is the local distance map and local direction map. The image and its segmentation annotation are from the COCO dataset (Lin et al., 2014)

Background. Locating accurate segmentation boundary is particularly important for medical images such as MRI, as the boundary for body organ is crucial. In a 2D image, we can represent the deviation of the predicted boundary with ground truth boundary in the form of a distance map, as shown in Figure 1. Each pixel in the example image is mapped with a normalized distance to its nearest ground truth boundary pixel, forming the boundary distance map.

For a typical pixel, the distance map could be reduced to a *local distance map* of 3×3, containing distance of the pixel itself and that of its eight neighbours. The cell with the smallest distance (*e.g.,* the top left cell in the example) represents the direction to the nearest boundary. To indicate this direction, local distance map can be further reduced to an one-hot *local direction map*, where the "1" cell representing the direction of the nearest boundary. Accordingly, the predicted boundary can be further refined toward this direction for more accurate boundary prediction (Wang et al., 2022). Span extraction tasks in NLP share the same aim to find accurate region boundaries, but in a 1D space,

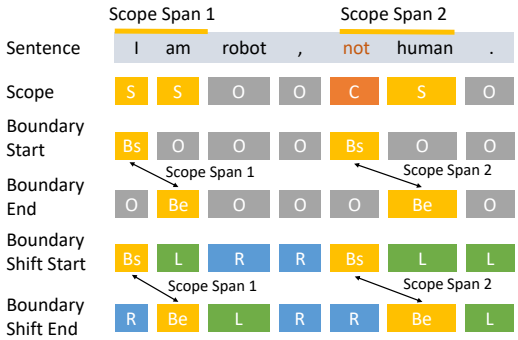


Figure 2: Example sentence labelling. Row 2 is the OSC scope tags. Rows 3 and 4 are "Boundary Start" (Bs) and "Boundary End" (Be) labels for scope spans. Rows 5 and 6 are Boundary Shift Map (BSM) labels, where L indicates left shift and R right shift.

i.e., along token sequence to shift left or right.

4.1 Boundary Shift Map

To enable boundary shift loss, we convert the scope labels to scope span boundary labels. $BS = \langle bs_1, bs_2, ..., bs_n \rangle$ and $BE = \langle be_1, be_2, ..., be_n \rangle$ are the two label sequences that represent the start and end of boundaries, respectively. bs_i is Bs if t_i is the start of a scope span, and O otherwise; be_i is Be if t_i is the end of a scope span, and O otherwise. If a span consists of only one token, the token itself is labeled both Bs and Be. Due to discontinuous spans, there could be multiple bs and be labels for one given cue, as shown in Figure 2.

Next, we create the "Boundary Shift Map" (BSM) for tokens that are not on the boundaries, by labeling their **shifting directions**: L for left, and R for right. The 5th and 6th rows in Figure 2 provide a visual illustration, for start and end boundaries respectively. A token is labeled with L / R if the nearest boundary resides on the left / right of the token. For the special case that a token has the same distance to both boundaries on the left and right, we label the token with R.

4.2 R-BSL Model Detail

Figure 3 illustrates the model architecture. We use BERT to encode the sentence and then use three feed-forward (FF) layers in parallel, to predict scope label and the BSM labels. The losses for the three label classifiers L_{scope}, L_{start}, L_{end} are the widely used Cross Entropy loss. L_{scope} is formally defined in Eq. 1 and the other two losses are defined similarly. The three losses are then combined to

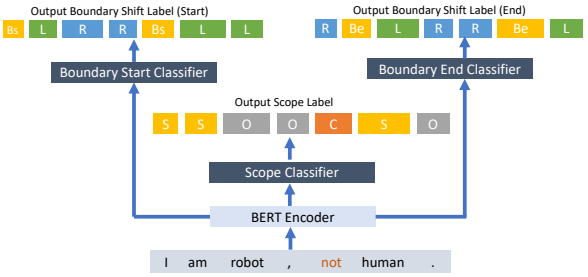


Figure 3: The model architecture.

form the final loss in Eq. 2, and we set $\alpha = 0.2$

$$L_{scope} = -\sum_{i=1}^{N} y^{(i)} \log(\hat{y}^{(i)}) \quad (1)$$

$$Loss = \alpha L_{scope} + \frac{1-\alpha}{2}(L_{start} + L_{end}) \quad (2)$$

Warm Up. In training, there is a "warm up" phase to train the model solely with scope loss L_{scope} for the first 5 epochs (where the validation loss is reasonably stable). Then boundary shift losses kick in to for scope refinement.

5 Experiments

5.1 Experiment Results

We conduct experiments on all three benchmark datasets: Sherlock, BioScope, and SFU. Among them, BioScope and SFU datasets do not come with official train-validation-test split. Following the previous studies, we use random split on 70-15-15 ratios; however the randomness in split may slightly affect model performance. Hence, we also report the result of our re-implemented baseline model Khandelwal and Sawant (2020), which is a BERT + Feed-forward with OSC scope tags.

Table 2 reports the results of F_1 over scope tokens, defined by Morante and Blanco (2012). For each scope, token-wise F_1 is computed between ground truth and predicted scope tokens. For all our implemented models, the reported results are average scores of 5 out of 7 runs, excluding the highest and lowest scores. All the runs are set with randomly generated seeds. Since Truong et al. (2022) use RoBERTa instead of BERT, we also report R-BSL (RoBERTa-base) for fair comparison.

R-BSL achieves best performance on all three datasets, particularly on Sherlock which comes with official train/test split. Note that on Sherlock dataset, our re-implemented baseline does not reach the scores reported in Khandelwal and

Dataset	Sherlock			BioScope-Abstract			SFU		
Method	Pr	Re	F_1	Pr	Re	F_1	Pr	Re	F_1
Khandelwal and Sawant, 2020	-	-	92.36	-	-	95.68	-	-	90.95
Khandelwal and Britto, 2020	-	-	-	-	-	96.68	-	-	**92.39**
Kurtz et al., 2020 *	-	-	89.71	-	-	-	-	-	-
Truong et al., 2022 ** - Baseline †	-	-	91.51	-	-	94.23	-	-	90.44
Truong et al., 2022 ** - CueNB	-	-	91.24	-	-	95.81	-	-	91.03
Baseline (Re-Implementation)	94.79	89.50	92.06	95.90	97.30	96.59	91.22	91.20	91.21
R-BSL (BERT-base-cased)	**95.12**	90.57	92.77	**96.33**	97.37	96.85	**91.55**	91.27	91.43
R-BSL (RoBERTa-base)	94.54	**91.24**	**92.85**	96.29	**98.54**	**97.40**	90.80	**91.51**	91.14

Table 2: Results are based on scope token level metrics. *Kurtz et al. (2020) does not use PLMs. **Truong et al. (2022) use RoBERTa-base instead of BERT-base. †The Baseline results are with the code released by Khandelwal and Sawant (2020). For BioScope-Abstract and SFU, there is no official train/test split. The difference in random split (with the same ratio) leads to the difference between our re-implemented baseline and previous studies.

Sawant (2020).[2] Truong et al. (2022) also reports lower results (mean of 5 runs) using the code released by Khandelwal and Sawant (2020). Nevertheless, both our R-BSL variants outperform all baselines on Sherlock, and on BioScope dataset. On SFU, our models' improvement is marginal. The main reason is the distributional bias, for the negation scopes largely align with punctuation or special tokens (see Appendix C).

For comprehensive evaluation, Table 3 shows the scope level F_1 scores by exact match. That is, when the predicted scope exactly matches the ground truth, it is considered as True Positive. There exists True Negative and False Positive cases due to "void negation" as discussed in Appendix C. When the ground-truth has no negation scope, if the model predicts any scope, that would be a False Positive. The scope exact match F_1 is similar to "Scope CM" metric defined in Morante and Blanco (2012). However, as we do not focus on cue detection but using cues as input, the results is not directly comparable with Scope CM results in earlier studies.

Compared to token-level measure, the improvements of our model over baseline is now by a much larger margin, particularly the variant with RoBERTa. In other words, the boundary refinement by BSL enables the model to resolve more accurate negation scopes in terms of exact scope span match, which is a stricter measure.

5.2 Ablation Study

We conduct two ablation studies on Sherlock dataset, and the results are reported in Table 4.

[2]The original paper does not provide complete experimental setup like how many runs were performed, or whether the reported results being mean or maximum of several runs.

Method	Sherlock	BioScope-A	SFU
Baseline (Re-Implemented)	84.19	94.11	88.06
R-BSL (BERT-base-cased)	85.35	94.94	88.50
R-BSL (RoBERTa-base)	**87.10**	**96.16**	**89.91**

Table 3: Scope exact match F_1 scores on three datasets

Model	Pr	Re	F_1
Baseline (Re-Implemented)	94.79	89.50	92.06
R-BSL (BERT-base-cased)	**95.12**	**90.57**	**92.77**
Replace BSL with Boundary Labels	95.34	89.27	92.10
Boundary classifier only	94.06	75.47	83.74
Without "warm up"	95.45	89.22	92.22

Table 4: Ablation studies on the Sherlock dataset

Boundary Label vs Boundary Shift Map. We first replace Boundary Shift Map with the start/end boundary labels (*i.e.,* Bs, Be, O tagging) as the prediction target for the boundary classifiers. Tiny improvement is observed over baseline, which indicates the usefulness of boundary labels. However, if not using scope classifier (or OSC tags) and using the boundary classifier with BSL loss, there is a significant drop in F_1. This result suggests that the scope span detection remains the key focus of the model and boundary classifier shall focus on boundary refinement.

"Warm Up" of Scope Classifier. We "warm up" the training with the first 5 epochs for scope classifier only. The boundary classifier with BSL loss then comes into the picture. To study its impact, we train all the three classifiers from the beginning. Shown in Table 4, the removal of warm up leads to negative impact on results. This ablation study suggests that the BSL can further improve the results when the span boundaries have been de-

tected by the base model, *i.e.,*, the scope classifier, at reasonably good accuracy.

6 Conclusion

We propose a simple sequence labelling training strategy to enhance boundary prediction for negation scope resolution. Through experiments, we demonstrate the effectiveness of boundary shift loss on complex span extraction tasks on three benchmark datasets. In particular, our simple model achieves the state-of-the-art results on the Sherlock dataset which is considered more challenging for this task. Our model is simple and can be used as a pre-processing for downstream tasks where negation is an important consideration.

Limitations

As shown in the ablation studies, using the Boundary Shift Loss without the base model for scope prediction leads to a huge negative impact on the performance. That is, BSL strongly relies on the assumption that the proposed candidate spans are, to some extent, being an accurate estimation of the target spans. The experiment of using BSL solely could be seen as an extreme case, that no candidate spans are proposed at all. For our task, BSL could benefit from the strong base model. For the case of noisy datasets or on a more challenging task, where a base model could not generalize to a reasonably good coarse span proposal, the benefit of BSL might be limited.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) Singapore, under its AME Programmatic Funding Scheme (Project #A19E2b0098).

References

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Valencia, Spain. Association for Computational Linguistics.

Aditya Khandelwal and Benita Kathleen Britto. 2020. Multitask learning of negation and speculation using transformers. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 79–87, Online. Association for Computational Linguistics.

Aditya Khandelwal and Suraj Sawant. 2020. NegBERT: A transfer learning approach for negation detection and scope resolution. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5739–5748, Marseille, France. European Language Resources Association.

Diederik P. Kingma and Jimmy Ba. 2015. 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*.

Natalia Konstantinova, Sheila C.M. de Sousa, Noa P. Cruz, Manuel J. Maña, Maite Taboada, and Ruslan Mitkov. 2012. A review corpus annotated for negation, speculation and their scope. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3190–3195, Istanbul, Turkey. European Language Resources Association (ELRA).

Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Roser Morante and Eduardo Blanco. 2012. *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), pages 265–274, Montréal, Canada. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Maite Taboada, Caroline Anthony, and Kimberly Voll. 2006. Methods for creating semantic orientation dictionaries. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Thinh Truong, Timothy Baldwin, Trevor Cohn, and Karin Verspoor. 2022. Improving negation detection with negation-focused pre-training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4188–4193, Seattle, United States. Association for Computational Linguistics.

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(11):1–9.

Chi Wang, Yunke Zhang, Miaomiao Cui, Peiran Ren, Yin Yang, Xuansong Xie, Xian-Sheng Hua, Hujun Bao, and Weiwei Xu. 2022. Active boundary loss for semantic segmentation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 2397–2405. AAAI Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

A Implementation Details

We use BERT-base-cased and RoBERTa-base as the pretrained LMs. The model parameters are optimized with Adam (Kingma and Ba, 2015). The BERT was trained with initial learning rate of $5e-6$, and the classifier layers were trained with initial learning rate of $5e-5$. This is different from Khandelwal and Sawant (2020) where they set both BERT and classifier layers learning rate to $5e-5$. The learning rate was scheduled with "Reduce Learning Rate on Plateau", which cuts the learning rate by half after 3 consecutive epochs without evaluation results being improved, and having cool-down of 2 epochs. We adopt early stopping threshold of 12, which means the training will be stopped when the evaluation results stop to improve for 12 consecutive epochs. The models were implemented with PyTorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2020).

Experiments were performed mainly on a single Nvidia RTX 3080 GPU. One training run took 40 minutes to 1 GPU hour, varied with the dataset size and early-stopping position. Inference time is 7 seconds on Sherlock test-set, 4 seconds on BioscopeAbstract test-set, 16 seconds on SFU test-set, and requires approximately 3 GB of GPU memory.

Following Khandelwal and Sawant (2020), we use special augmentation tokens to indicate the appearance of negation cues. However, our indicator tokens are slightly different from Khandelwal and Sawant (2020) where they only add one special token in front. We have special tokens on both ends of a [cue] cue [/cue]. Another small difference is the treating of affixal cues. The Sherlock dataset defines affixal cues like 'in-" being cue and "-frequent" being inside the scope for word "infrequent". In our implementation, we simply treat the whole word as a cue and use post-processing to handle the affixal cue by regular expressions. This is also for the sake of unifying the model behaviour

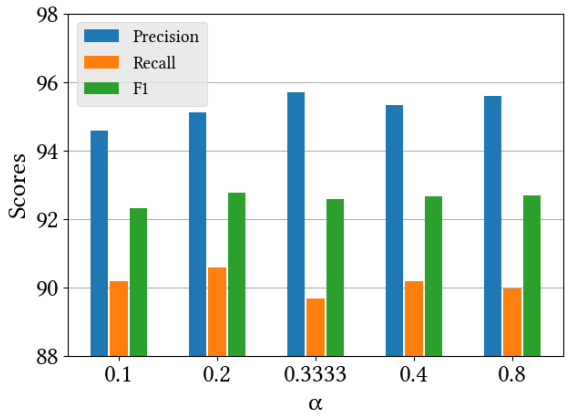


Figure 4: Token level metrics on Sherlock dataset with different α.

across datasets, as the SFU and BioScope datasets do not have such special annotation for affixal cues.

All three datasets contain not only sentences with negations, but contain sentences without negations, for training model on predicting negation cues. As our focus is on negation scope resolution with the assumption that cue has been detected, we only use the portion of sentences containing negations for training and testing.

B Impact of α in Training Loss

We perform hyper-parameter search on the value of α in the loss function (Equation 2). As shown in Figure 4, it appears that the Precision improves as α increase; the Recall improves with α initially at lower range, but decreases as α further increase. The "trade-off" between Precision and Recall is best balanced at $\alpha = 0.2$, where we observe the highest F_1. Note that $\alpha = 0.3333$ denotes the case that no weighting terms are applied, *i.e.*, $Loss = L_{scope} + L_{start} + L_{end}$. It is equivalent to $\alpha = 0.3333$ though no α was actually applied in implementation, for the sake of stability of floating point calculations. Overall, the impact of α on token level F_1 score was not significant, since the boundary shift losses serve as auxiliary loss, and the final prediction is still based on scope classifier.

C Discussion on Distributional Bias

Comparing the token-level F_1 (Table 2) and the span-level exact match F_1 (Table 3), both F_1's are similar on BioScope and SFU dataset, but not on the Sherlock dataset. As Fancellu et al. (2017) had suggested, the annotation rules of BioScope and SFU had seemingly over-simplified the problem. They also provided statistics on percentages of negation scopes that can be exactly represented by the closest punctuation tokens to the cue as scope boundaries. The percentage for BioScope Abstracts and SFU are 64% and 80%, respectively, while the value for Sherlock is only 40%.

Number of scopes and spans	Sherlock	BioScope-A	SFU
Total number of negation scopes	1,421	1,719	3,528
Number of discontinuous scopes	134	15	0
Total number of scope spans	1,571	1,735	3,068
Scope spans that exactly wrapped by punctuation and special tokens	744 (47.4%)	1,373 (79.1%)	2,577 (84.0%)

Table 5: Statistics on scope boundaries and punctuation and special tokens

However, for BERT-based models, researchers often rely on using special tokens (*e.g.,* [cue] cue [/cue] for a cue) as indicators for region of interest. The special tokens themselves can be considered as another form of punctuation tokens. Here we provide another set of statistics on scope spans that can be exactly represented by punctuation tokens and cue special tokens, in Table 5. Note that in Sherlock and SFU, the negation cue tokens are not considered as part of scope in their annotation. This would cause considerably number of additional discontinuous negation scopes, hence we adjust the annotation when performing this statistics to also consider cue tokens as part of scope. Also for SFU, the number of scopes (3528) is much higher than the number of scope spans (3068), as there are a good number of "void" scope. These are the cases that the negation in the sentence is the cue itself, such as "Of course **not**!". If the annotation rule does not consider negation cues as part of negation scope, there will be no negation scope in such sentences, and hence we call them "void negation". Such cases are not considered as negation in annotation guideline of BioScope.

Reported in Table 5, the percentage of scope spans that are exactly wrapped by punctuation (considering special tokens also as punctuation) for BioScope dataset is 79.1%, and for SFU 84.0%. Such phenomenon could be due to both the annotation scheme, and the writing style. Such high percentage of "easy cases" could make the model biased to relying more on punctuation information, and yet deliver relatively high scores. In the mean time, the percentage for Sherlock is 47.4%, and the increase of percentage due to cue special token is far less than that of BioScope.

The high percentage values also explain that the exact match F_1's of BioScope and SFU are quite close to their token-level F_1 scores. The Sherlock dataset, hence is considered as a more challenging dataset for this problem.

While one would intuitively think of re-sampling the datasets to adjust the portion of easy and hard cases, Fancellu et al. (2017) show that the benefit of under-sampling is marginal on their LSTM-based models. We presume similar behaviour for our BERT-based model.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations (After 6. Conclusion)

☒ A2. Did you discuss any potential risks of your work?
This is a task on information extraction from given text. There is no potential risk.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1. Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

2. Related Work, 5. Experiments

☑ B1. Did you cite the creators of artifacts you used?
2. Related Work, 5. Experiments

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
2. Related Work, 5. Experiments

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
2. Related Work, 5. Experiments

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Among all three datasets used, two are anonymized by nature (English novel published more than hundred years ago, and scientific papers abstracts), one was anonymized by the creators.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
2. Related Work, 5. Experiments

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix C

C ☑ Did you run computational experiments?

5. Experiments

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5. Experiments, Appendix A, Appendix B

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5. Experiments

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
5. Experiments, Appendix A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.