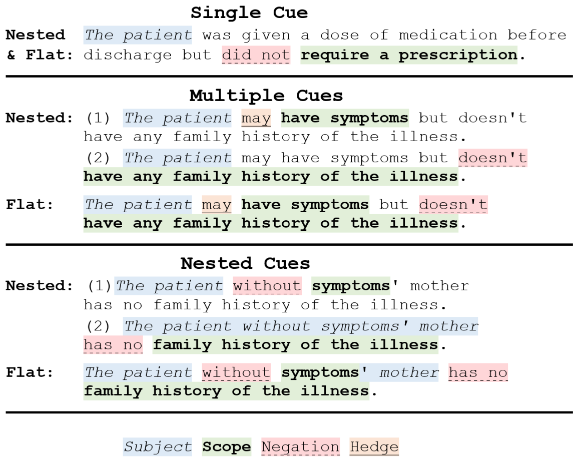
Flat and Nested Negation and Uncertainty Detection with PubMed BERT

Anonymous ACL submission

Abstract

Negation and uncertainty detection is an oft-studied challenge in biomedical NLP. Annotation style for the task has not been standardized and as such, the existing datasets not only vary in domain but require various algorithmic designs due to their structural differences. We present a new negation detection dataset in two versions from clinical publications. We further developed two BERT-based models to evaluate on each dataset version. Both models treat the task as a token-level multi-class classification task, one of which is capable of assigning more than one label per token in the case of recursive nesting. Our models achieve F1 scores of 76% and 72% on the development and test sets, respectively.



```
                    Single Cue
Nested   The patient was given a dose of medication before
& Flat:  discharge but did not require a prescription.

                    Multiple Cues
Nested:  (1) The patient may have symptoms but doesn't
         have any family history of the illness.
         (2) The patient may have symptoms but doesn't
         have any family history of the illness.
Flat:    The patient may have symptoms but doesn't
         have any family history of the illness.

                    Nested Cues
Nested:  (1)The patient without symptoms' mother
         has no family history of the illness.
         (2) The patient without symptoms' mother
         has no family history of the illness.
Flat:    The patient without symptoms' mother has no
         family history of the illness.

              Subject Scope Negation Hedge
```

Figure 1: The trained flat and nested models' outputs when a fed single-cue sentence, a multiple-cue sentence, and a nested-cue sentence.

1 Introduction

Negation and uncertainty detection is of particular importance in the biomedical NLP domain. While benchmark datasets exist, there are few, and they generally all follow different annotations schemes, or do not fall within the biomedical domain. Recent approaches to the task treat it as a token-level, multi-class classification problem (where each word is assigned one and only one label). As human language is recursive, the task of identifying negation and speculation cues along with their relevant scopes often involves nesting, wherein a single word can have multiple labels depending on which cue it interacts with. Likewise, sentences containing multiple cues can have nested or overlapping scopes, and resolving the structure of which cues and scopes to group together is crucial. When treating negation detection as a token-level classification task, the ability to capture nesting or information regarding structural relationships between cues and their scopes can be lost. We make two contributions: First, we created a new clinical text dataset annotated for negation detection in two variations, a flat version and nested version. Second, we developed two models to evaluate on the datasets, one designed specifically to capture nesting and structural information between cues and scopes.

2 Related Work

A common approach to negation and uncertainty detection involves a two-step process in which negation and speculation cue words are first identified, and then scope resolution is performed. Uncertainty detection is sometimes referred to as 'modality' or 'speculation' detection, cue words are those which directly express the negation or uncertainty (e.g. 'not', 'possibly'), and scope is defined as the part of the sentence affected by the negation or uncertainty cue.

Interest in negation detection has largely centered around its use in information retrieval and extraction in clinical texts. An early approach employed a rule-based algorithm which first identifies UMLS terms (Bodenreider, 2004), then extracts negation cues from a pre-defined list and finally greedily selects surrounding words using regular expressions for scope resolution (Chapman, Bridewell, Hanbury, Cooper, and Buchanan, 2001). Later work framed negation detection as a token-level classification task (Morante, Liekens, and Daelemans, 2008). Subsequent work proceeded to focus more on scope resolution than cue detection and employed deep learning architecture (Qian, Li, Zhu, Zhou, Luo, and Luo, 2016). With the advent of transfer learning, more recent and SOTA methods involve using a BERT encoder (Devlin et al., 2019). The authors of NegBERT train two BERT encoders separately for the task of cue detection and scope resolution (Khandelwal and Sawant, 2020). This model is adapted to instead employ multitask learning whereby the same BERT encoder is trained for cue detection and scope resolution (Khandelwal and Britto, 2020).

A variety of datasets exist for the task of negation detection, and no standard annotation method yet exists. The NegEx dataset (Chapman et al., 2001) is clinicial and only annotates conditions which can be experienced by a person, and also labels if the condition happened recently or not. For example, the sentence 'Extremities reveal no peripheral cyanosis or edema.' would label 'cyanosis' and 'edema' as 'negated, recent, patient'. The i2b2 2010 dataset only annotates whether problem mentions are positive or negated (Uzuner, South, Shen, and DuVall, 2011). The BioScope corpus annotates only cues and scopes, where subjects are not included in the scope. For example, where bold indicates a cue and underline indicates a scope, the BioScope corpus would annotate the following sentence as: 'The man did**n't** see the woman.' The ConanDoyle-neg dataset is similar to the BioScope dataset, except it includes the subject in the scope and additionally annotates the main event in the scope being negated or questioned. For example: 'The man did**n't** *see* the woman,' where *see* is the event. The latter two datasets additionally label only the negation affixes when negation cues appear as such, e.g. '**un**able'.

3 Data

We collected data from PubMed abstracts, resulting in 3252 sentences. We reserved 10% of the dataset for development set and 10% for a test set. Tables 1 and 2 provide further statistics on our resulting datasets. Additionally, we make our dataset publicly available through the HuggingFace dataset library[1] as 'pubmed_neg'.

	#	% full	% ann.
sentences	3252	-	-
w/ 1+ cue	879	27%	-
w/ hedge	483	15%	55%
w/ negation	464	14%	53%
w/ 2+ cues	161	5%	18%

Table 1: Flat Dataset Statistics. '% full' refers to the percentage of the entire dataset, '% ann.' refers to the percentage of the data containing one or more negation cues.

	#	% full	% ann.
sentences	3252	-	-
w/ 1+ cue	877	27%	-
w/ hedge	490	15%	55%
w/ negation	491	15%	56%
w/ 2+ cues	228	7%	26%
w/ nesting	78	2%	9%

Table 2: Nested Dataset Statistics. '% full' refers to the percentage of the entire dataset, '% ann.' refers to the percentage of the data containing one or more negation cues.

The sentences were first annotated by a linguist, and then by five Amazon Mechanical Turk[2] workers per-sentence. Mechanical Turk workers were compensated $0.05-$0.06 per sentence, including those containing neither negation nor uncertainty cues. The annotations were consolidated such that the linguists' annotations were given equal weight to the entirety of the MTurk annotations. We performed annotation consolidation automatically by implementing the approach described by Amazon SageMaker Ground Truth for its Named Entity Recognition annotation consolidation[3]: "Named entity recognition clusters text selections by Jaccard similarity and calculates selection boundaries based on the mode, or the median if the mode isn't clear. The label resolves to the most assigned entity label in the cluster, breaking ties by random selection."

[1] https://huggingface.co/datasets

[2] https://www.mturk.com/

[3] https://docs.aws.amazon.com/sagemaker/latest/dg/sms-annotation-consolidation.html, accessed 09.09.2021

3.1 Annotation Guidelines

Annotators were instructed to first determine whether a negation or uncertainty cue was present in a sentence. If so, to then identify the subject and scope of the cue. We define the subject as the noun phrase (possibly) lacking the action, concept, item, etc. which is being negated or questioned. The scope, by contrast, is the action, concept, item, etc. whose existence is negated or questioned. The cue is then the entire verb phrase up until the specific word indicating negation or uncertainty, e.g. 'He [did not]$_{\text{NEG}}$ go', Annotators were instructed to ignore nesting, and annotate only the outermost negation/speculation. For example, '[The man without the hat]$_{\text{SUBJ}}$ [did not]$_{\text{NEG}}$ [see the dog]$_{\text{SCOPE}}$.' Further, words containing affixed cues were annotated as a whole as the cue, such that the whole word 'unable' was labeled as the negation cue. Lastly, negation-affixed words which denote or belong to a discrete medical concept (e.g. 'antibodies', 'progression-free survival') were not to be annotated.

3.2 Post-Processing

Following consolidation, an additional linguist manually reviewed the annotations and made adjustments where the consolidation produced a noisy, incorrect output as a result of very diverse annotator inputs; annotators missed a cue; or annotators incorrectly labeled something as a cue (e.g. 'antibodies'). The linguist added further information disambiguating relationships between cues and subjects/scopes when multiple discrete cues occur in a single sentence, or when nesting occurs (i.e. spans embedded in other spans). This produced an additional dataset where all cues are annotated, including nested ones. We therefore present two versions of the dataset: a flat and a nested one.

4 Methods

4.1 Flat Model

Our flat model is a token-level multi-class classifier implemented with the HuggingFace Transformer library (Wolf et al., 2020). The classifier receives contextualized word vectors from PubMed BERT[4] (Gu et al., 2020) and determines to which of the 6 classes a token belongs: Subject, Scope, Negation cue, Hedge cue, None, or padding. A hedge cue is an uncertainty/speculation cue.

4.2 Nested Model

Our nested model is also a token-level multi-class classification task identical to the flat model except where specified. The model consists of a BERT encoder and two learned classifiers: a Cue Detection Classifier and a Scope Detection Classifier. The model first identifies cues in a sentence, and then the scope and subject of each cue separately. Once cues are identified by the Cue Detection Classifier, discrete cues are identified as any contiguous sequence of the same cue label (e.g. if 3 words in a row are labeled as a negation cue, this is taken to represent a single negation cue). During training, the Cue Detection Classifier's gold labels are used for this. Then, for each identified cue, we either *insert* a special token immediately before and after an identified cue span, or *replace* each identified cue token with this special token. Both methods have been explored with success by Khandelwal and Sawant (2020) and the insertion method by Khandelwal and Britto (2020). The modified sentences are passed again to the same BERT encoder, and the output is passed to the Scope Detection Classifier, where each token is determined to be a scope or subject of the cue span in question. The losses from the Cue Detection and Scope Detection Classifiers are summed before performing backpropagation, thereby employing multi-task learning.

This nested approach preserves the multi-class classification structure while inherently allowing words to have multiple labels (as is the case with nesting). Additionally, the output contains information about which cue belongs to which subject and scope when multiple cues are present in a single sentence. Neither of these are supported by our Flat Model.

4.3 Evaluation

We use the F1 metric to evaluate the performance of our models. We compute precision as $\frac{\#correct}{\#predicted}$, recall as $\frac{\#correct}{\#gold}$, and F1 as $\frac{2PR}{P+R}$. Our implementation is token-level and does not consider a correctly assigned 'No Label' as a correct prediction, which we feel would inflate precision. Because a token can have more than one label in the nested scenario, we consider two ways to compute correct predictions for precision and recall: a flat and a nested method. The nested method counts how

[4] https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext

Data	Model	Nested			Flat		
		F1	P	R	F1	P	R
Flat$_D$	Flat	**76.1**	80.2	72.5	76.1	80.2	72.5
	Nested	72.9	71.4	**74.5**	75.6	76.8	74.5
Nested$_D$	Flat	73.4	**80.6**	67.4	75.7	**80.6**	71.3
	Nested	74.3	75.6	73.0	**76.7**	78.7	**74.7**
Flat$_T$	Flat	70.3	73.5	67.3	70.3	73.5	67.3
	Nested	68.4	67.1	**69.9**	71.8	73.8	**69.9**
Nested$_T$	Flat	70.3	73.5	67.3	70.3	73.5	67.3
	Nested	**71.5**	**74.4**	68.9	**73.4**	**78.6**	68.8

Table 3: Top models' performances on the development$_D$ and test$_T$ sets for **all labels** for each dataset version.

Data	Model	Nested			Flat		
		F1	P	R	F1	P	R
Flat$_D$	Flat	70.7	72.4	69.1	70.7	72.4	69.1
	Nested	68.1	65.5	**71.1**	68.1	65.5	71.1
Nested$_D$	Flat	67.9	**73.8**	62.9	68.4	**73.8**	63.7
	Nested	**71.6**	72.7	70.6	**72.1**	72.7	**71.4**
Flat$_T$	Flat	71.6	77.0	67.0	71.6	77.0	67.0
	Nested	69.9	72.5	**67.4**	69.9	72.5	**67.4**
Nested$_T$	Flat	71.6	77.0	67.0	71.6	77.0	67.0
	Nested	**73.3**	**81.0**	66.9	**73.5**	**81.0**	67.2

Table 4: Top models' performances on the development$_D$ and test$_T$ sets for **cue labels only** for each dataset version.

Data	Model	Nested			Flat		
		F1	P	R	F1	P	R
Flat$_D$	Flat	**76.7**	81.1	72.9	76.7	81.1	72.9
	Nested	73.4	72.0	**74.8**	76.5	78.2	74.8
Nested$_D$	Flat	74.0	**81.4**	67.9	76.5	**81.4**	72.2
	Nested	74.5	75.9	73.2	**77.2**	79.5	**75.1**
Flat$_T$	Flat	70.1	73.1	67.4	70.1	73.1	67.4
	Nested	68.3	66.4	**70.2**	72.0	74.0	**70.2**
Nested$_T$	Flat	70.1	73.1	67.4	70.1	73.1	67.4
	Nested	**71.3**	**73.6**	69.1	**73.4**	**78.3**	69.1

Table 5: Top models' performances on the development$_D$ and test$_T$ sets for **subject and scope labels only** for each dataset version.

many times a label is assigned to each token in the gold labels, and the number of times that label was assigned to that token in the predictions, and takes the overlap as the number of correct predictions. If a token is assigned the Subject label 3 times, '#predicted' += 3. The flat method disregards the number of times a label was assigned to a given token, and just considers a correct prediction if a label was assigned to a token at all. If a token is assigned the Subject label 3 times, '#predicted' += 1, but if a token is assigned the Subject label 3 times and the Scope label once, '#predicted' += 2. The nested method harshly penalizes precision when the nested model predicts too many cues, and recall when the model predicts too few.

5 Results

Tables 3, 4, and 5 show the model performances on each dataset, according to each metric. The metrics reported for each model are from a single model (not an average). Each model was trained on its respective dataset, and then evaluated on both. Both model architectures perform better on their respective dataset, in regards to overall performance (Table 3) and cue (Table 4) and scope (Table 5) detection. The nested model achieves the highest F1 score on the (nested) test set, and generally outperforms the flat model. Both models perform worse on the test data than on the development data, likely due to overfitting.

5.1 Discussion

Not only does the nested model achieve the highest F1 score on the test data, it returns useful information regarding groupings of subject, cue, and scope. This is particularly useful when sentences contain multiple negations or hedges. The utility of this can be seen in the model outputs illustrated in Figure 1. Further, the unconventional annotation choice to label entire negation-affixed words such as 'unable' as a negation cue was supported by the fact that PubMed BERT does not tokenize this word into the sub-word tokens 'un' and '##able'. As such, our models would simply not be capable of assigning a negation cue label to only the prefix 'un-'.

6 Conclusion

We created two datasets for the task of negation and uncertainty detection in clinical publications: a flat and a nested version. We further developed two models to evaluate on the datasets as well as two metrics to account for nested labels. Of our two models, the nested model is able to capture not only token-level labels, but also the which subject, scope, and cues are related to one another. This is particularly useful when sentences contain multiple cues or some form of recursive nesting. Finally, we publicly release both versions of our dataset.

References

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database-Issue):267–270.

Wendy Chapman, Will Bridewell, Paul Hanbury, Gregory Cooper, and Bruce Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34:301–310.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Aditya Khandelwal and Benita Kathleen Britto. 2020. Multitask learning of negation and speculation using transformers. In *LOUHI*.

Aditya Khandelwal and Suraj Sawant. 2020. Negbert: A transfer learning approach for negation detection and scope resolution.

Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the scope of negation in biomedical texts. pages 715–724.

Zhong Qian, Peifeng Li, Qiaoming Zhu, Guodong Zhou, Zhunchen Luo, and W. Luo. 2016. Speculation and negation scope detection via convolutional neural networks. In *EMNLP*.

Ozlem Uzuner, Brett South, Shuying Shen, and Scott DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA*, 18:552–6.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.