

Detecting the scope of negations in clinical notes

Giuseppe Attardi, Vittoria Cozza, Daniele Sartiano

Dipartimento di Informatica - Universit  di Pisa

Largo B. Pontecorvo, 3

{attardi, cozza, sartiano}@di.unipi.it

Abstract

English. We address the problem of automatically detecting the scope of negations and speculations in clinical notes, by proposing a machine-learning algorithm that analyzes the dependency tree of a sentence. Given a negative/speculative cue, the algorithm tries to extend the boundary of the scope towards the left and the right, by navigating through the parse tree. We report on experiments with the algorithm using the Bioscope corpus.

Italiano. *Il lavoro affronta il problema di identificare l'ambito a cui si applica una negazione o un'espressione dubitativa nel testo di un referto medico. Si propone un algoritmo di apprendimento automatico, che analizza l'albero di parsing di ogni frase. Dato un indizio di negazione/ipotesi, l'algoritmo cerca di estendere il confine dell'ambito sia a destra che a sinistra, attraversando l'albero di parsing. Riportiamo infine i risultati di esperimenti con l'algoritmo effettuati usando il corpus Bioscope.*

1 Introduction

Clinical notes are a vast potential source of information for healthcare systems, from whose analysis valuable data can be extracted for clinical data mining tasks, for example confirming or rejecting a diagnosis, predicting drug risks or estimating the effectiveness of treatments. Clinical notes are written in informal natural language, where, besides annotating evidence collected during a patient visit, physician report historical facts about the patient and suggested or discarded hypothesis. Annotations about dismissed hypotheses or evidence about the absence of a phenomenon are particularly abundant in these notes and should be recognized as such in order to avoid misleading conclusions. A standard keyword based search engine might for ex-

ample return many irrelevant documents where a certain symptom is mentioned but it does not affect the patient.

Medical records are currently analysed by clinical experts, who read and annotate them manually. In some countries like Spain, it has become mandatory by law for all medical records to be annotated with the mentions of any relevant reported fact, associated with their official ICD9 code. To assign the right ICD9 code, it is of critical importance to recognize the kind of context of each mention: assertive, negative or speculative. In the BioScope corpus, a collection of bio-medical text, one out of eight sentences indeed contains negations (Vincze et al. (2008)).

In order to automate the process of annotation of clinical notes, the following steps can be envisaged:

1. recognition of medical entities, by exploiting techniques of named entity (NE);
2. normalization and association to a unique official concept identifier to their key terminology from UMLS metathesaurus (O. Bodenreider, 2004);
3. detection of negative or speculative scope.

NE recognition and normalization steps can be performed by relying on shallow analysis of texts (for an exhaustive and updated overview of the state of the art, see Pradhan et al. (2014)). The identification of negative or speculative scope, instead, cannot just rely on such simple text analysis techniques, and would require identifying relations between parts, by means of a deeper syntactic-semantic analysis of sentences.

This work presents a novel algorithm that learns to determine the boundaries of negative and speculative scopes, by navigating the parse tree of a sentence and by exploiting machine learning techniques that rely on features extracted from the analysis of the parse tree.

2 Related Work

Negation and uncertainty detection are hard issues for NLP techniques and are receiving in-

creasing attention in recent years. For the detection of negative and speculative scope, both rule-based approaches and machine learning approaches have been proposed.

Harkema et al. (2010) propose a rule-based algorithm for identifying trigger terms indicating whether a clinical condition is negated or deemed possible, and for determining which text falls within the scope of those terms. They use an extended cue lexicon of medical conditions (Chapman et al., 2013). They perform their analysis for English as well as for low resources languages, i.e., Swedish. Their experiments show that lexical cues and contextual features are quite relevant for relation extraction i.e., negation and temporal status from clinical reports.

Morante et al. (2008) explored machine-learning techniques for scope detection. Their system consists of two classifiers, one that decides which tokens in a sentence are negation signals, and another that finds the full scope of these negation signals. On the Bioscope corpus, the first classifier achieves an F1 score of 94.40% and the second 80.99%.

Also Daz et al. (2012) propose a two-stage approach: first, a binary classifier decides whether each token in a sentence is a negation/speculation signal or not. A second classifier is trained to determine, at the sentence level, which tokens are affected by the signals previously identified. The system was trained and evaluated on the clinical texts of the BioScope corpus. In the signal detection task, the classifier achieved an F1 score of 97.3% in negation recognition and 94.9% in speculation recognition. In the scope detection task, a token was correctly classified if it had been properly identified as being inside or outside the scope of all the negation signals present in the sentence. They achieved an F1 score of 93.2% in negation and 80.9% in speculation scope detection.

Sohn et al. (2012) developed hand crafted

rules representing subtrees of dependency parsers of negated sentences and showed that they were effective on a dataset from their institution.

Zou et al. (2015) developed a system for detecting negation in clinical narratives, based on dependency parse trees. The process involves a first step of negative cue identification that exploits a binary classifier. The second step instead analyses the parse tree of each sentence and tries to identify possible candidates for a negative scope extracted with a heuristics: starting from a cue, all ancestors of the cue are considered, from which both the full subtree rooted in the ancestor and the list of its children are considered as candidates. A classifier is then trained to recognize whether any of these candidates falls within the scope of the cue. The system was trained on a Chinese corpus manually annotated including scientific literature and financial articles. At prediction time, besides the classifier, also a set of rules based on a suitable lexicon is used to filter the candidates and to assign them to the scope of a cue. Since the classifier operates independently on each candidate, it may happen that a set of discontinuous candidates is selected. A final clean up step is hence applied to combine them. This system achieved an F1 score below 60%.

3 Negation and speculation detection

For the cue negation/speculation detection, we apply a sequence tagger classifier that recognizes phrases annotated with negation and speculation tags. The cui exploits morphological features, attribute and dictionary features.

For scope detection, we implemented a novel algorithm that explores the parse tree of the sentence, as detailed in the following.

3.1 Scope Detection

For identifying negative/speculative contexts in clinical reports, we exploit information from the

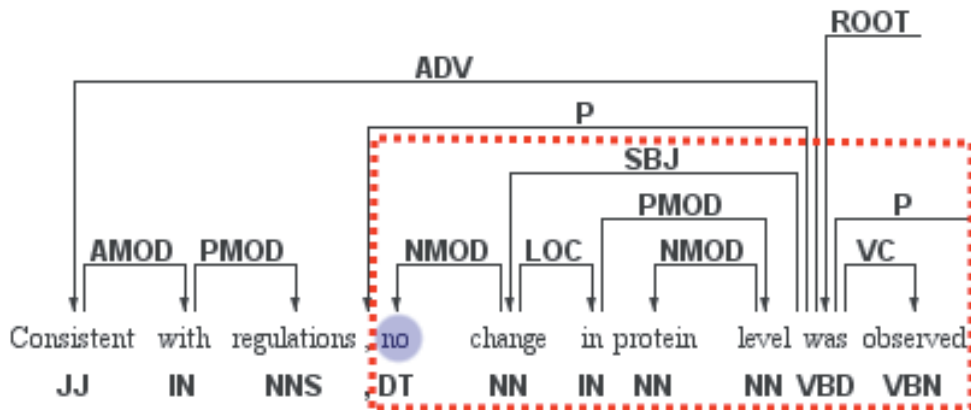


Figure 0. Example of parse tree with a negative scope.

parse tree of sentences. Our approach is however different from the one by Zou et al. (2015), which has the drawback, as mentioned earlier, of operating independently on subtrees and hence it requires an extra filtering step to recombine the candidates and to exclude poor ones according to lexical knowledge.

Our approach assumes that scopes are contiguous and they contain the cue. Hence, instead of assembling candidates independently of each other, our process starts from a cue and tries to expand it as far as possible with contiguous subtrees either towards the left or towards the right.

In the description of the algorithm, we will use the following definitions.

Definition. *Scope adjacency order* is a partial order such that, for two nodes x, y of a parse tree, $x < y$ iff x and y are consecutive children of the same parent, or x is the last left child of y or y is the first right child of x .

Definition. *Right adjacency list.* Given a word w_i in a parse tree, the right adjacency list of w_i ($RAL(w_i)$) consists of the union of $RA = \{w_j \mid w_i < w_j\}$ plus $RAL(y)$ where y is the node in RA with the largest index.

Definition. *Left adjacency list.* Symmetrical of Left adjacency list.

The algorithm for computing the scope S of a cue token at position c in the sentence, exploits the definitions of RAL and LAL and is described below.

Algorithm.

1. $S = \{w_c\}$
2. for w_i in $LAL(w_c)$ sorted by reverse index
 - if w_i belongs to the scope,
 - $S = S \cup \{w_k \mid i \leq k < c\}$
 - otherwise proceed to next step.
3. for w_i in $RAL(w_c)$ sorted by index
 - if w_i belongs to the scope,
 - $S = S \cup \{w_k \mid c < k \leq i\}$
 - Otherwise stop.

In essence, the algorithm moves first towards the left as far as possible, and whenever it adds a node in step 2, it also adds all its right children, in order to ensure that the scope remains contiguous. It then repeats the same process towards the right.

Lemma. *Assuming that the parse tree of the sentence is non-projective, the algorithm produces a scope S consisting of consecutive tokens of the sentence.*

The proof descends from the properties of non-projective trees.

The decision on whether a candidate belongs to a scope is entrusted to a binary classifier which is trained on the corpus, using features from the nodes in the context of the candidate.

These are nodes selected from the parse tree. In particular, there will be two cases to consider, depending on the current step of the algorithm. For example, in step 2 the nodes considered are illustrated in Figure 1.

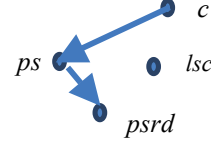


Figure 1. lsc is the leftmost child of c within the current scope, ps is its left sibling, $psrd$ is the rightmost descendant of ps .

Below we show which nodes are considered for feature extraction in step 3:

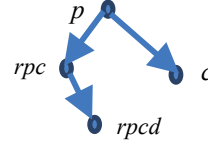


Figure 2. c is the leftmost child of p , rpc is its rightmost child of p , $rpcd$ is the rightmost descendant of rpc .

The features extracted from these tokens are: form, lemma, POS, dependency relation type of the candidate node c , the cue node, $rpcd$ and $psrd$; the distance between node c and the cue node; the number of nodes in the current scope; if there are other cues in the subtree of node c ; the dependency relation types of the children of node c ; whether the nodes $psrd$ and $rpcd$ are within the scope; the part of speech, form, lemma and dependency relation types of lsc and rpc .

We illustrate which nodes the algorithm would visit, on the parse tree of Figure 0. The negative cue is given by the token `no`, marked in grey in the figure. Initially $S = \{\text{no}\}$, and $LAL(\text{no}) = \{\text{Consistent}, , \}$, while $RAL(\text{no}) = \{\text{change}, \text{in}, \text{level}, \text{was}, \text{observed}, .\}$. The word with largest index in LAL is `.`, it is not within the scope, hence S stays the same and we proceed to step 3. The token with smallest index in RAL is `change`, which is part of the scope, hence $S = \{\text{no}, \text{change}\}$. The next token is `in`, which also gets added to S , becoming $S = \{\text{no}, \text{change}, \text{in}\}$. The next token is `level`, which is part of the scope: it is added to the scope as well as all

tokens preceding it (protein), obtaining {no, change, in, protein, level}. The next two tokens are also added and the algorithm terminates when reaching the final dot, which is not part of the scope, producing $S = \{\text{no, change, in, protein, level, was, observed}\}$.

Lemma. *The algorithm always terminates with a contiguous sequence of tokens in S that include the cue.*

Notice that differently from (Zou et al. (2015)), our algorithm may produce a scope that is not made of complete subtrees of nodes.

3.2 Experiments

We report an experimental evaluation of our approach on the BioScope corpus, where, according to Szarvas et al. (2008), the speculative or negative cue is always part of the scope.

We pre-processed a subset of the corpus for a total of 17.766 sentences, with the Tanl pipeline (Attardi et al., 2009a), then we splitted it into train, development and test sets of respectively 11.370, 2.842 and 3.554 sentences.

In order to prepare the training corpus, the BioScope corpus was pre-processed as follows. We applied the Tanl linguistic pipeline in order to split the documents into sentences and to perform tokenization according to the Penn Treebank (Taylor et al., 2003) conventions. Then POS tagging was performed and finally dependency parsing with the Desr parser (Attardi, 2006) trained on the GENIA Corpus (Kim et al. 2003).

The annotations from BioScope were integrated back into the pre-processed format using an IOB notation (Speranza, 2009). In particular, two extra columns were added to the CoNLL-X file format. One column for representing negative or speculative cues, using tags NEG and SPEC along with a cue id. One other column for the scope, containing the id of the cue it refers to, or `_i` if the token is not within a scope. If a token is part of more then one scope, the id of the cue of each scope is listed, separated by comma.

Here is an example of annotated sentence:

ID	FORM	CUE	SCOPES
1	The	O	—
2	results	O	—
3	indicate	B-SPEC	3
4	that	I-SPEC	3
5	expression	O	3
6	of	O	3
7	these	O	3
8	genes	O	3

9	could	B-SPEC	3, 9
10	contribute	O	3, 9
11	to	O	3, 9
12	nuclear	O	3, 9
13	signaling	O	3, 9
15	mechanisms	O	3, 9

where could contribute to nuclear signaling mechanisms is a nested scope within indicate that expression of these genes could contribute to nuclear signaling mechanisms, whose cues are respectively could and indicate that.

For the cue detection task, we experimented with three classifiers:

1. a linear SVM classifier implemented using the libLinear library (Fan et al. 2008)
2. Tanl NER (Attardi et al., 2009b), a statistical sequence labeller that implements a Conditional Markov Model.
3. deepNL (Attardi, 2015) is a Python library for Natural Language Processing tasks based on a Deep Learning neural network architecture. DeepNL also provides code for creating word embeddings from text using either the Language Model approach by Collobert et al. (2011) or Hellinger PCA, as in (Lebret et al., 2014).

The features provided to classifiers 1) and 2) included morphological features, lexical features (i.e. part of speech, form, lemma of the token and its neighbours), and a gazetteer consisting of all the cue words present in the training set.

The solution based on DeepNL reduces the burden of feature selection since it uses word embeddings as features, which can be learned through unsupervised techniques from plain text; in the experiments, we exploited the word embedding from Collobert et al. (2011). Besides word embeddings, also discrete features are used: suffixes, capitalization, Part of speech and presence in a gazetteer extracted from the training set.

The best results achieved on the test set, with the above mentioned classifier, are reported in Table 1.

	Precision	Recall	F1
LibLinear	88.82%	90.46%	89.63%
Tanl NER	91.15%	90.31%	90.73%
DeepNL	88.31%	90.69%	89.49%

Table 1. Negation/Speculation cue detection results.

The classifier, used in the algorithm of scope detection for deciding whether a candidate be-

longs to a scope or not, is a binary classifier, implemented using libLinear.

The performance of the scope detection algorithm is measured also in terms of Percentage of Correct Scopes (PCS), a measure that considers a predicted scope correct if it matches exactly the correct scope. Precision/Recall are more tolerant measures since they count each correct token individually.

The results achieved on our test set from the BioScope corpus are reported in Table 2.

Precision	Recall	F1	PCS
78.57%	79.16%	78.87%	54.23%

Table 2. Negation/Speculation Scope detection results

We evaluated the performance of our algorithm also on the dataset from the CoNLL 2010 task 2 and we report the results in Table 3, compared with the best results achieved at the challenge (Morante et al. 2010).

	Precision	Recall	F1
Morante et al.	59.62%	55.18%	57.32%
Our system	61.35%	63.68%	62.49%

Table 3. Speculation scope detection

We can note a significant improvement in Recall, that leads also to an relevant improvement in F1.

4 Conclusions

We have described a two-step approach to speculation and negation detection. The scope detection step exploits the structure of sentences as represented by its dependency parse tree. The novelty with respect to previous approaches also exploiting dependency parses is that the tree is used as a guide in the choice of how to extend the current scope. This avoids producing spurious scopes, for example discontinuous ones. The algorithm also may gather partial subtrees of the parse. This provides more resilience and flexibility. The accuracy of the algorithm of course depends on the accuracy of the dependency parser, both in the production of the training corpus and in the analysis. We used a fast transition-based dependency parser trained on the Genia corpus, which turned out to be adequate for the task. Indeed in experiments on the BioScope corpus the algorithm achieved accuracy scores above the state of the art.

References

Giuseppe Attardi. 2006. Experiments with a Multilingual Non-Projective Dependency Parser, *Proc.*

of the Tenth Conference on Natural Language Learning, New York, (NY).

Giuseppe Attardi et al.. 2009a. Tanl (Text Analytics and Natural Language Processing). SemaWiki project: <http://medialab.di.unipi.it/wiki/SemaWiki>

Giuseppe Attardi, et al. 2009b. The Tanl Named Entity Recognizer at Evalita 2009. In *Proc. of Workshop Evalita 09 - Evaluation of NLP and Speech Tools for Italian*, Reggio Emilia, ISBN 978-88-903581-1-1.

Giuseppe Attardi. 2015. DeepNL: a Deep Learning NLP pipeline. *Workshop on Vector Space Modeling for NLP, NAACL 2015*, Denver, Colorado (June 5, 2015).

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, vol. 32, no. supplement 1, D267 D270.

Wendy W. Chapman, Dieter Hilert, Sumithra Velupillai, Maria Kvist, Maria Skeppstedt, Brian E. Chapman, Michael Conway, Melissa Tharp, Danielle L. Mowery, Louise Deleger. 2013. Extending the NegEx Lexicon for Multiple Languages. *Proceedings of the 14th World Congress on Medical & Health Informatics (MEDINFO 2013)*.

Ronan Collobert et al. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, 2461 2505.

N. P. Cruz D az, et al. 2012. A machine-learning approach to negation and speculation detection in clinical texts. *Journal of the American society for information science and technology*, 63.7, 1398 1410.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9(2008), 1871 1874.

Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. 2010. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, Volume 42, Issue 5, 839 851.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun-ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-text mining. *ISMB (Supplement of Bioinformatics)*, pp. 180 182.

R mi Lebre and Ronan Collobert. 2014. Word Embeddings through Hellinger PCA. *EACL 2014*: 482.

Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the scope of negation in biomedical texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Proc-*

- essing (EMNLP '08). Association for Computational Linguistics, Stroudsburg, PA, USA, 715-724.
- Roser Morante, Vincent Van Asch, and Walter Daelemans. 2010. Memory-based resolution of in-sentence scopes of hedge cues. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning Shared Task*. Association for Computational Linguistics, 2010.
- Sameer Pradhan, et al. 2014. SemEval-2014 Task 7: Analysis of Clinical Text. *Proc. of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, August 2014, Dublin, Ireland, pp. 5462.
- Sunghwan Sohn, Stephen Wu, Christopher G. Chute. 2012. Dependency parser-based negation detection in clinical narratives. *Proceedings of AMIA Summits on Translational Science*. 2012: 1.
- Maria Grazia Speranza. 2009. The named entity recognition task at evalita 2007. *Proceedings of the Workshop Evalita*. Reggio Emilia, Italy.
- Gyrgy Szarvas, Veronika Vincze, Richard Farkas, Janos Csirik, The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts, *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, June 19-19, 2008, Columbus, Ohio.
- Ann Taylor, Mitchell Marcus and Beatrice Santorini. 2003. The Penn Treebank: An Overview, chapter from Treebanks, *Text, Speech and Language Technology*, Volume 20, pp 5-22, Springer Netherlands.
- Veronika Vincze, Gyrgy Szarvas, Richard Farkas, Gyrgy Mra, Janos Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, 9(Suppl 11), S9.
- Bowei Zou, Guodong Zhou and Qiaoming Zhu. Negation and Speculation Identification in Chinese Language. *Proceeding of the Annual ACL Conference 2015*.