

# Learning the scope of negation in biomedical texts

Blind

## Abstract

In this paper we present a machine learning system that finds the scope of negation in biomedical texts. The system consists of two memory-based engines, one that decides if the tokens in a sentence are negation signals, and another that finds the full scope of the negation signals. Our approach to negation detection differs in two main aspects from existing research. First, we focus on finding the scope of negation signals, instead of determining whether a term is negated or not. Second, we apply supervised machine learning techniques, whereas most existing systems apply rule-based algorithms. As far as we know, this way of approaching the negation scope finding task is novel.

## 1 Introduction

In this paper we present a machine learning system that finds the scope of negation in biomedical texts. The system consists of two memory-based engines, one that decides if the tokens in a sentence are negation signals (i.e., words indicating negation), and another that finds the full scope of these negation signals. Finding the scope of a negation signal means determining at sentence level which words in the sentence are affected by the negation. Our approach differs in two main aspects from existing research. First, we focus on finding the scope of negation signals, instead of determining whether a term is negated or not. Second, we apply supervised machine learning techniques, whereas most existing systems apply rule-based algorithms.

Predicting the scope of negation is important in information extraction from text for obvious reasons; instead of simply flagging the sentences containing negation as not suited for extraction (which is currently the best that can be done), correct semantic relations can be extracted when the scope of negation is known, providing a better recall.

Not being able to recognize negation can also hinder automated indexing systems (Mutalik et al., 2001; Rokach et al., 2008). As Mutalik et al. (2001) put it, “to increase the utility of concept indexing of medical documents, it is necessary to record whether the concept has been negated or not”. They highlight the need to detect negations in examples like “no evidence of fracture”, so that an information retrieval system does not return irrelevant reports.

A main goal of this research is to find out to which extent a machine learning system can perform the scope finding task. In order to do that, we choose a memory-based learning algorithm. Memory-based language processing (Daelemans and van den Bosch, 2005) belongs to the class of lazy (as opposed to eager) learning methods, descendants of k-*nn* statistical pattern matching. It is based on the idea that NLP problems can be solved by reuse of solved examples of the problem in memory, applying similarity-based reasoning on these examples in order to solve new problems. As language processing tasks typically involve many subregularities and (pockets of) exceptions, it has been argued that lazy learning is at an advantage in solving these highly disjunctive learning problems compared to eager learning, as the latter eliminate not only noise but also potentially useful exceptions (Daelemans et al.,

1999). Memory-based algorithms have been successfully applied in language processing to a wide range of linguistic tasks, ranging from phonology to semantic analysis. In this paper we show that memory-based learning can also be applied to predicting the scope of negation.

The paper is organised as follows. In Section 2 we summarise related work. In Section 3, we describe the corpus with which the system has been trained. In Section 4 we introduce the task to be performed by the system, which is described in Section 5. The results are presented and discussed in Section 6. Finally, Section 7 puts forward some conclusions.

## 2 Related work

Most research regarding negation in the biomedical domain focuses on detecting if a medical term is negated or not, whereas in this paper we focus on detecting the full scope of negation signals.

Chapman et al. (2001) developed NegEx, a regular expression based algorithm for determining whether a finding or disease mentioned within narrative medical reports is present or absent. The reported results are 94.51 precision and 77.84 recall.

Mutalik et al. (2001) developed Negfinder, a rule-based system that recognises negated patterns in medical documents. It consists of two tools: a lexical scanner called lexer that uses regular expressions to generate a finite state machine, and a parser that relies on LALR grammars. The reported results using independent negation detection are 95.70 recall and 91.80 precision.

Sanchez-Graillet and Poesio (2007) present an analysis of negated interactions in biological texts and an heuristic-based system that extracts such information. They treat all types of negation: (i) Affixal negation, which is expressed by an affix. (ii) Noun phrase or emphatic negation, expressed syntactically by using a negative determiner (e.g. *no*, *nothing*). (iii) Inherent negation, expressed by words with an inherently negative meaning (e.g. *absent*). (iv) Negation with explicit negative particles (e.g. *no*, *not*). The texts are 50 journal articles. The preliminary results reported range from 54.32 F-score to 76.68, depending on the method applied.

Elkin et al. (2005) describe a rule-based system that assigns to concepts a level of certainty as part

of the generation of a two-phase dyadic parse tree. First a preprocessor breaks each sentence into text and operators. Then, a rule base decides if a concept has been positively, negatively, or uncertainly asserted. The system achieves 97.20 recall and 98.80 precision.

The systems mentioned above are essentially based on lexical information. Huang and Lowe (2007) propose a classification scheme of negations based on syntactic categories and patterns in order to locate negated concepts, regardless of their distance from the negation signal. Their hybrid system that combines regular expression matching with grammatical parsing achieves 92.60 recall and 99.80 precision.

Additionally, Boytcheva et al. (2005) incorporate the treatment of negation in a system, MEHR, that extracts from electronic health records all the information required to generate automatically patient's chronicles. According to the authors "the negation treatment module inserts markers in the text for negated phrases and determines scope of negation by using negation rules". However, in the paper there is no description of the rules that are used and there is no explanation about how the results presented for negation recognition (57% of negations correctly recognised) are evaluated.

Most of the above-mentioned research applies rule-based algorithms to negation finding. Machine learning techniques have been applied in some cases. Averbuch et al. (2004) developed an algorithm to learn automatically negative context patterns that uses information gain.

Golding and Chapman (2003) apply machine learning techniques to distinguish whether a medical observation is negated by the word *not*. Their corpus contains 207 selected sentences from hospital reports, in which a negation appears. They use Naive Bayes and Decision Trees and achieve a maximum of 90 F-score. According to the authors, their main finding is that "when negation of a UMLS term is triggered with the negation phrase *not*, if the term is preceded by *the* then do not negate".

Goryachev et al. (2006) compare the performance of four different methods of negation detection, two regular expression-based methods and two classification-based methods trained on 1745 discharge reports. They show that the regular

expression-based methods have better agreement with humans and better accuracy than the classification methods. Like in most of the mentioned work, the task consists in determining if a medical term is negated.

Rokach et al. (2008) present a new pattern-based algorithm for indentifying context in free-text medical narratives. The originality of the algorithm lies in that it automatically learns patterns similar to the manually written patterns for negation detection.

Apart from work on finding if a term is negated or not, we are not aware of research that has focused on learning the full scope of negation signals. This is why we consider that the research presented in this paper provides a new approach to the treatment of negation scope in natural language processing.

### 3 Corpus

The corpus used is a part of the BioScope corpus (Szarvas et al., 2008)<sup>1</sup>, a freely available resource that consists of medical and biological texts. Every sentence is annotated with information about negation and speculation, namely the boundaries of the scope and the keywords, as shown in (1).

- (1) PMA treatment, and <xscope id="X1.4.1"><cue type="negation" ref="X1.4.1">not<cue> retinoic acid treatment of the U937 cells</xscope> acts in inducing NF-KB expression in the nuclei.

A first characteristic of the annotation of scope in the BioScope corpus is that it is based on linguistic principles: all sentences that assert the non-existence or uncertainty of something are annotated, in contrast to other corpora where only sentences containing biomedical terms are annotated. A second characteristic is that the annotation is extended to the biggest syntactic unit possible.

The part used in our experiments are the biological paper abstracts from the GENIA corpus (Collier et al., 1999). This part consists of 1,273 abstracts in 11,872 sentences. We discarded five sentences due to annotation errors. The total number of words used is 313,222, from which 1,739 are negation signals that belong to the different types as in described in (Sanchez-Graillet and Poesio, 2007).

We processed the texts with the GENIA tagger (Tsuruoka and Tsujii, 2005; Tsuruoka et al.,

2005), a bidirectional inference based tagger that analyzes English sentences and outputs the base forms, part-of-speech tags, chunk tags, and named entity tags in a tab-separated format. Additionally, we converted the annotation about scope of negation into a token-per-token representation.

Table 1 shows an example sentence of the corpus that results from converting and processing the BioScope representation. Like in the recent CoNLL Shared Tasks, sentences are separated by a blank line and fields are separated by a single tab character. A sentence consists of tokens, each one starting on a new line. A token consists of the following 10 fields:

1. ABSTRACT ID: number of the GENIA abstract.
2. SENTENCE ID: sentence counter starting at 1 for each new abstract.
3. TOKEN ID: token counter, starting at 1 for each new sentence.
4. FORM: word form or punctuation symbol.
5. LEMMA: lemma of word form.
6. POS TAG: Penn Treebank part-of-speech tags described in (Santorini, 1990).
7. CHUNK TAG: IOB chunk tag produced by the GENIA tagger.
8. NE TAG: IOB named entity tags produced by the GENIA tagger.
9. NEG CUE: tokens that are negation signals are marked as NEG. Negation signals in the BioScope corpus are single words, except for the signal *could not*. After the tagging process the signal *cannot* becomes also multiword because the tagger splits it in two words. In these cases we assign the NEG mark to *not*.
10. NEG SCOPE: IOB tags that indicate if a token is at the beginning of the negation scope (B-NEG), in the remaining (I-NEG), or out (O-NEG). These tags have been obtained by transforming the xml files of BioScope. Each token can have one or more NEG SCOPE tags, depending on the number of negation signals in

<sup>1</sup>Web page: [www.inf.u-szeged.hu/rgai/bioscope](http://www.inf.u-szeged.hu/rgai/bioscope).

10415075	07	1	NF-kappa	NF-kappa	NN	B-NP	B-protein	-	B-NEG O-NEG
10415075	07	2	B	B	NN	I-NP	I-protein	-	I-NEG O-NEG
10415075	07	3	binding	binding	NN	I-NP	O	-	I-NEG O-NEG
10415075	07	4	activity	activity	NN	I-NP	O	-	I-NEG O-NEG
10415075	07	5	was	be	VBD	B-VP	O	-	I-NEG O-NEG
10415075	07	6	absent	absent	JJ	B-ADJP	O	NEG	I-NEG O-NEG
10415075	07	7	in	in	IN	B-PP	O	-	I-NEG O-NEG
10415075	07	8	several	several	JJ	B-NP	O	-	I-NEG O-NEG
10415075	07	9	SLE	SLE	NN	I-NP	O	-	I-NEG O-NEG
10415075	07	10	patients	patient	NNS	I-NP	O	-	I-NEG O-NEG
10415075	07	11	who	who	WP	B-NP	O	-	I-NEG O-NEG
10415075	07	12	were	be	VBD	B-VP	O	-	I-NEG O-NEG
10415075	07	13	not	not	RB	I-VP	O	NEG	I-NEG B-NEG
10415075	07	14	receiving	receive	VBG	I-VP	O	-	I-NEG I-NEG
10415075	07	15	any	any	DT	B-NP	O	-	I-NEG I-NEG
10415075	07	16	medication	medication	NN	I-NP	O	-	I-NEG I-NEG
10415075	07	17	,	,	,	O	O	-	I-NEG I-NEG
10415075	07	18	including	include	VBG	B-PP	O	-	I-NEG I-NEG
10415075	07	19	corticosteroids	corticosteroid	NNS	B-NP	O	-	I-NEG I-NEG
10415075	07	20	.	.	.	O	O	-	O-NEG O-NEG

Table 1: Example sentence of the corpus BioScope converted into columns format

the sentences. The tags are separated by a blank space and appear in the order of appearance of the negation signals in the sentence.

#### 4 Task description

We approach the scope finding task as a classification task that consists of classifying the tokens of a sentence as being a negation signal or not, and as being inside or outside the scope of the negation signal(s) (as many times as there are negation signals in the sentence). The information that can be used to train the system is in columns 1 to 8 of Table 1. The information to be predicted by the system is in columns 9 and 10.

As far as we know, this way of approaching the negation scope finding task is novel, whereas at the same time it conforms to the well established standards of the recent CoNLL Shared Tasks<sup>2</sup>. By setting up the task in this way we show that the negation scope finding task can be modelled in a way similar to semantic role labeling.

<sup>2</sup>Web page of CoNLL:  
<http://www.ifarm.nl/signll/conll/>.

## 5 System description

In order to solve the task, we apply supervised machine learning techniques. We build a memory-based scope finder, that tackles the task in two phases. In the first phase a classifier predicts if a token is a negation signal, and in the second phase another classifier predicts if a token is inside the scope of each of the negation signals. The output of the classifier is postprocessed applying a sequence optimisation algorithm.

We use the IB1 classifier as implemented in TiMBL (version 6.1.2) (Daelemans et al., 2007), a supervised inductive algorithm for learning classification tasks based on the  $k$ -nearest neighbor classification rule (Cover and Hart, 1967). In IB1, similarity is defined by computing (weighted) overlap of the feature values of a test instance and a memorized example. The metric combines a per-feature value distance metric with global feature weights that account for relative differences in discriminative power of the features.

### 5.1 Negation signal finding

In this phase, a classifier predicts if a word is a negation signal or not. The IB1 algorithm was parameterised by using overlap as the similarity metric, gain

ratio for feature weighting, using 7  $k$ -nearest neighbors. All neighbors have equal weight when voting for a class. The instances represent all tokens in the corpus and they have the following features:

- About the word: Word form, lemma, part of speech, and chunk IOB tag.
- About the word context: Word form, POS, and IOB tag of the three previous and three next words. To obtain the previous word we perform a linear left-to-right search. This is how *previous* has to be interpreted further on when features are described.

## 5.2 Scope finding

In the first step of this phase, a classifier predicts if a token is in the scope of each of the negation signals of a sentence. The instances represent a combination of a negation signal and a token. This means that all tokens in a sentence are combined with as many negation signals as there are in the sentence. For example, token *NF-kappa* in Table 1 will be represented in two instances as shown in (2):

- (2) NF-kappa absent [features] B-NEG  
NF-kappa not [features] O-NEG

Negation signals are those that have been classified as such in the previous phase. Only sentences that have negation signals are selected for this phase.

The IB1 algorithm was parameterised by using overlap as the similarity metric, gain ratio for feature weighting, using 7  $k$ -nearest neighbors, and weighting the class vote of neighbors as a function of their inverse linear distance.

The features of the scope finding classifier are:

- About the negation signal: word form, POS, chunk IOB tag, type of chunk (NP, VP, ...), and word form, POS, chunk IOB tag, type of chunk, and named entity of the 3 previous and 3 next words.
- About the combining word: word form, POS, chunk IOB tag, type of chunk, named entity, and word form, POS, chunk IOB tag, type of chunk, and named entity type of the 3 previous and 3 next words.

- About the tokens between the negation signal and the word: Chain of POS types, distance in number of tokens, and chain of chunk IOB tags.
- Others: Binary feature indicating if the word and the negation signal are in the same chunk. Location of the word relative to the negation signal (pre, post, same).

In the second step of this phase, we apply a sequence optimisation algorithm in order to increase the number of fully correct scopes. A scope is fully correct if all tokens in a sentence have been assigned their correct scope label for a given negation signal. The scope finding classifier can make predictions that result in non-consecutive blocks of scope tokens, whereas negation signals scope always over a consecutive block of tokens. We observed that in 54.05% of the cases the predicted scopes of negation were non-consecutive when running the system with gold-standard negation signals. The function of the algorithm is to output consecutive blocks of scope tokens.

The algorithm that we apply checks first if the predicted signal is indeed in the scope of the negation. If the signal is predicted to be out of the scope, the algorithm overwrites the predicted scope annotation in order to include the signal in its scope. Given the position of the signal in the sentence, the algorithm determines the starting and ending tokens of the consecutive block of predicted scope tokens that surrounds the signal. Other blocks of predicted scope tokens may still be predicted outside of this block, but they are separated from the current scope, which contains the signal, by tokens that have been predicted not to be in the scope of the negation. If the tokens that have been predicted to be out of scope are identified as noise (i.e. potentially wrong predictions), the algorithm should overwrite their annotation and link separated blocks of scope together. Apart from that, if the loose blocks of predicted scope appear to be noise, the algorithm should overwrite their annotation marking them as being out of scope.

Consider a sentence where the cue is in one block of predicted scope of length  $k$  tokens. Suppose now that there is another block of  $m$  consecutive tokens that is predicted as scope, but that is separated from

the main scope block by  $l$  tokens which are predicted to be out of scope. If non-consecutive blocks are near each other, i.e., if the number  $l$  of out-of-scope tokens is sufficiently small in comparison with the numbers  $k$  and  $m$  of scope tokens, then the intermediate tokens that have been predicted out of scope could be considered as noise. In contrast, if there are too many intermediate tokens that separate two blocks of scope, then the additional block of scope is probably wrongly annotated. Following this logic, if  $l < \alpha(k + m)$ , with a specifically chosen  $\alpha$ , the intermediate out-of-scope tokens are re-annotated as scope tokens, and the separated blocks are connected to form one bigger block containing the negation signal. Otherwise, the loose block of scope is re-annotated to be out of scope. When the main scope is extended, and more blocks are found that are separated from the main scope block, the algorithm reiterates this procedure until one consecutive block of scope tokens has been found. Dependent on whether blocks need to be added before or after the main scope block, we have observed in preliminary tests that  $\alpha = 0.2$  for extending the main scope block backward, and  $\alpha = 0.3$  for extending the block further forward into the sentence provide the best results.

## 6 Results

The results of the systems have been obtained by performing 10-fold cross validation experiments. The evaluation is made using the precision and recall measures (Van Rijsbergen, 1979), and their harmonic mean, F-Measure. We calculate micro F1. In the negation finding task, we consider that a negation token is correctly classified if it has been assigned a NEG class. In the scope finding task, we consider that a token is correctly classified if all the IOB tag(s) that it has been assigned are correct. This means that when there is more than one negation signal in the sentence, the token has to be correctly assigned an IOB tag for as many negation signals as there are. For example, token NF-kappa from Table 1 reproduced in (3) will not be correct if it is assigned classes I-NEG O-NEG or B-NEG I-NEG.

- (3) 10415075 07 1 NF-kappa NF-kappa NN B-NP  
B-protein \_ B-NEG O-NEG

Additionally, we evaluated the percentage of fully correct scopes (PCS).

### 6.1 Negation signal finding

We calculate two baselines for negation signal finding. Baseline 1 (B1) is calculated by assigning the NEG class to all the tokens that had *no* or *not* as lemma. The F1 of the baseline is 80.66. Baseline 2 (B2) is calculated by assigning the NEG class to all the tokens that had *no*, *not*, *lack*, *neither*, *unable*, *without*, *fail*, *absence*, or *nor* as lemma. These lemmas account for 85.85 % of the negation signals.

Baseline	Total	Prec.	Recall	F1
B1	1739	90.42	72.80	80.66
B2	1739	89.77	93.38	91.54

Table 2: Baselines of the negation finding system.

Table 3 shows the results of the negation signal finding system. With F1 94.40, it outperforms Baseline 2 by 2.86 points.

Neg signals	Total	Prec.	Recall	F1
lack	88	100.00	100.00	100.00
neither	42	100.00	100.00	100.00
unable	30	100.00	100.00	100.00
without	83	100.00	98.79	99.39
nor	44	100.00	100.00	98.89
rather	19	95.00	100.00	97.43
not	1057	96.15	96.97	96.56
no	209	95.73	96.65	96.18
none	7	85.71	85.71	85.71
fail	57	79.36	87.71	83.33
miss	2	66.66	100.00	80.00
absence	57	67.64	80.70	73.60
failure	8	45.54	62.50	52.63
absent	13	42.85	23.07	30.00
could	6	66.66	33.33	44.44
either	2	0.00	0.00	0.00
impossible	1	0.00	0.00	0.00
instead	3	0.00	0.00	0.00
lacking	1	0.00	0.00	0.00
loss	1	0.00	0.00	0.00
negative	1	0.00	0.00	0.00
never	2	0.00	0.00	0.00
or	1	0.00	0.00	0.00
with	6	0.00	0.00	0.00
Overall	1739	94.21	94.59	94.40

Table 3: F scores of the negation finding classifier.

Precision and recall are very similar. Scores show a clear unbalance between different negation signals.

Those with the lowest frequency are difficult to find. Probably, this could be avoided by training the system with a bigger corpus. Those with the highest frequency are identified successfully.

*Could* appears as a negation signal because the tagger does not assign to it the lemma *can*, but *could*, causing the wrong assignment of the tag NEG to *not*, instead of *could* when the negation cue in BioScope is *could not*. This bug will be fixed in the next version.

## 6.2 Scope finding

We provide the results of the classifier and the results of applying the scope sequence optimisation algorithm to the output of the classifier.

Table 4 shows results for two versions of the scope finding classifier, one based on gold standard negation signals (GS NEG), and another (PR NEG) based on negation signals predicted by the classifier described in the previous section.

	<b>Prec.</b>	<b>Recall</b>	<b>F1</b>	<b>PCS</b>
GS NEG	85.86	85.42	85.63	36.28
PR NEG	80.13	78.03	79.06	34.87

Table 4: Results of the scope finding classifier with gold-standard (GS NEG) and with predicted negation signals (PR NEG).

The F1 of PR NEG is 6.57 points lower than the F1 of GS NEG, which is an expected effect. Precision and recall of GS NEG are very balanced, whereas PR NEG has a lower recall than precision. These measures are the result of a token per token evaluation, which does not guarantee that the complete sequence of scope is correct. This is reflected in the low percentage of fully correct scopes of both versions of the classifier.

In Table 5, we present the results of the system after applying the scope sequence optimisation algorithm.

	<b>Prec.</b>	<b>Recall</b>	<b>F1</b>	<b>PCS</b>
GS NEG	88.61	77.89	82.92	55.49
PR NEG	80.94	81.59	81.26	50.80

Table 5: Results of the system with gold-standard (GS NEG) and with predicted negation signals (PR NEG) after applying the scope sequence optimisation algorithm.

<b>Scope tags</b>	<b>Total</b>	<b>Prec.</b>	<b>Recall</b>	<b>F1</b>
O-NEG	29590	86.77	85.96	86.36
B-NEG O-NEG O-NEG	3	75.00	100.00	85.71
B-NEG	1423	82.48	84.04	83.25
O-NEG O-NEG O-NEG	46	100.00	81.50	81.22
I-NEG	11567	72.97	78.36	75.57
O-NEG O-NEG	2848	83.51	68.15	75.05
B-NEG O-NEG	108	70.43	75.00	72.64
B-NEG B-NEG	32	100.00	56.25	72.00
I-NEG O-NEG O-NEG	17	54.83	100.00	70.83
I-NEG I-NEG	638	62.44	65.67	64.01
O-NEG B-NEG	100	78.12	50.00	60.97
O-NEG I-NEG	691	68.89	50.65	58.38
I-NEG O-NEG	884	42.41	66.06	51.65
B-NEG B-NEG O-NEG	3	100.00	33.33	50.00
O-NEG I-NEG I-NEG	37	100.00	18.91	31.81
I-NEG I-NEG O-NEG	66	42.00	24.00	31.00
I-NEG I-NEG I-NEG	20	26.92	35.00	30.43
I-NEG B-NEG	13	14.00	53.84	22.22
B-NEG I-NEG	1	0.00	0.00	0.00
I-NEG B-NEG B-NEG	1	0.00	0.00	0.00
I-NEG I-NEG B-NEG	1	0.00	0.00	0.00
O-NEG B-NEG B-NEG	2	0.00	0.00	0.00
O-NEG O-NEG B-NEG	2	0.00	0.00	0.00
O-NEG O-NEG I-NEG	12	0.00	0.00	0.00
Overall	48105	80.94	81.59	81.26

Table 6: F scores and percentage of correct scopes (PCS) of the system after applying the scope sequence optimisation algorithm.

The most remarkable result is the 30.15 and 24.43 error reduction in the percentage of fully correct scopes of GS NEG and PR NEG respectively, which shows that the algorithm is efficient for this corpus.

Also interesting is the increase in precision of GS NEG. Aiming at high precision in negation finding systems is desirable, since extracting as negated information that is not negated can have negative effects, especially in the biomedical domain, where information extraction systems might be used to fill in medical reports about patients. However, applying the algorithm causes a decrease in recall of 7.53 points. The application of the algorithm provokes a trade-off between precision and recall. However, we consider that having a higher percentage of correct scopes is more important than having a higher F1 in a token per token evaluation.

As for PR NEG, the application of the algorithm causes a moderate increase in recall, while keeping the precision at a very similar level.

Table 6 shows detailed results of the system based on predicted negation signals after applying the

scope sequence optimisation algorithm. Classes O-NEG, B-NEG, and I-NEG are among the most frequent and get high scores. Classes that are composed of three tags are more difficult to predict with scores under the 50 F1, except for O-NEG O-NEG O-NEG. Less frequent classes tend to be less predictable.

Neg signals	Total	Correct	PCS
without	82	58	70.73
no	211	137	64.92
rather	20	12	60.00
not	1066	569	53.37
neither	42	19	45.23
none	7	3	42.85
lack	87	31	35.63
fail	63	21	33.33
missing	3	1	33.33
absent	7	2	28.57
nor	43	11	25.58
absence	68	17	25.00
unable	30	6	20.00
failure	11	0	0.00
could	3	0	0.00
negative	1	0	0.00
never	1	0	0.00
box	1	0	0.00
Overall	1746	887	50.80

Table 7: Information about Percentage of Correct Scopes (PCS) per negation signal.

Table 7 shows information about the percentage of correct scopes per negation signal after applying the algorithm. The negation signals listed are the ones predicted by the classifier that predicts negation signals. This explains the presence of *box* in the list, which is an incorrect prediction.

The signal with the highest percentage is *without*, followed by *no*, *rather* and *not*, which are above 50%. It would be interesting to investigate how the syntactic properties of the negation signals are related to the percentage of correct scopes, and how does the algorithm perform depending on the type of signal. Nouns and adjectives expressing negation tend to have a lower percentage than adverbs.

## 7 Conclusions

In this paper we have presented a machine learning system that finds the scope of negation in biomedical

texts. The system consists of two memory-based engines, one that decides if the tokens in a sentence are negation signals, and another that finds the full scope of the negation signals. The first engine achieves 94.40 F1, and the second 79.06. However, the evaluation in terms of correct scopes shows the weakness of the system. This is why a sequence optimisation algorithm is applied. The algorithm achieves an error reduction of 24.43, with 50.80 % of fully correct scopes. Although there is room for improvement, the results suggest that machine learning techniques are suited for tackling the task.

We have highlighted the fact that our approach to negation detection focuses on finding the scope of negation signals, instead of determining whether a term is negated or not, and on applying supervised machine learning techniques. As far as we know, this approach is novel.

Additionally, we have shown that negation finding can be modelled for natural language processing purposes in a way similar to other linguistic tasks like semantic role labeling. In our model, scope finding is represented as a classification task that consists of classifying the tokens of a sentence as being a negation signal or not, and as being inside or outside the scope of the negation signal(s) (as many times as there are negation signals in the sentence).

Further research should be carried on in several directions. In the first place, the fact that most negation scopes coincide with boundaries of chunks suggests that scope could be predicted chunk per chunk instead of token per token. Secondly, other machine learning algorithms should be integrated in the system in order to optimise performance. Thirdly, the system should be tested in different types of biomedical texts, like full papers or medical reports to check if the system is robust. Fourthly, research about sequence optimisation could lead to improvements in the algorithm. Finally, the data should be qualitatively analysed in order to find explanations for the upper boundaries of the system.

## Acknowledgments

Blind.



## References

- M. Averbuch, T. Karson, B. Ben-Ami, O. Maimon, and L. Rokach. 2004. Context-sensitive medical information retrieval. In *Proceedings of the 11th World Congress on Medical Informatics (MEDINFO-2004)*, pages 1–8, San Francisco, CA. IOS Press.
- S. Boytcheva, A. Strupchanska, E. Paskaleva, and D. Tcharaktchiev. 2005. Some aspects of negation processing in electronic health records. In *Proceedings of International Workshop Language and Speech Infrastructure for Information Access in the Balkan Countries*, pages 1–8, Borovets, Bulgaria.
- W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B.G. Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*, 34:301–310.
- N. Collier, H.S. Park, N. Ogata, Y. Tateisi, C. Nobata, T. Sekimizu, H. Imai, and J. Tsujii. 1999. The GENIA project: corpus-based knowledge acquisition and information extraction from genome research papers. In *Proceedings of EACL-99*.
- T. M. Cover and P. E. Hart. 1967. Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27.
- W. Daelemans and A. van den Bosch. 2005. *Memory-based language processing*. Cambridge University Press, Cambridge, UK.
- W. Daelemans, A. Van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning, Special issue on Natural Language Learning*, 34:11–41.
- W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch. 2007. TiMBL: Tilburg memory based learner, version 6.1, reference guide. Technical Report Series 07-07, ILK, Tilburg, The Netherlands.
- P. L. Elkin, S. H. Brown, B. A. Bauer, C.S. Husser, W. Carruth, L.R. Bergstrom, and D. L. Wahner-Roedler. 2005. A controlled trial of automated classification of negation from clinical notes. *BMC Medical Informatics and Decision Making*, 5(13).
- I. M. Goldin and W.W. Chapman. 2003. Learning to detect negation with ‘Not’ in medical texts. In *Proceedings of ACM-SIGIR 2003*.
- S. Goryachev, M. Sordo, Q.T. Zeng, and L. Ngo. 2006. Implementation and evaluation of four different methods of negation detection. Technical report, DSG.
- Y. Huang and H.J. Lowe. 2007. A novel hybrid approach to automated negation detection in clinical radiology reports. *J Am Med Inform Assoc*, 14(3):304–311.
- A.G. Mutalik, A. Deshpande, and P.M. Nadkarni. 2001. Use of general-purpose negation detection to augment concept indexing of medical documents. a quantitative study using the UMLS. *J Am Med Inform Assoc*, 8(6):598–609.
- L. Rokach, R. Romano, and O. Maimon. 2008. Negation recognition in medical narrative reports. *Information Retrieval Online*.
- O. Sanchez-Graillet and M. Poesio. 2007. Negation of protein-protein interactions: analysis and extraction. *Bioinformatics*, 23(13):424–432.
- B. Santorini. 1990. Part-of-speech tagging guidelines for the penn treebank project. Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.
- G. Szarvas, V. Vincze, R. Farkas, and J. Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of BioNLP 2008: Current Trends in Biomedical Natural Language Processing*, pages 38–45, Columbus, Ohio, USA. ACL.
- Y. Tsuruoka and J. Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of HLT/EMNLP 2005*, pages 467–474.
- Y. Tsuruoka, Y. Tateishi, J. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, 2005. *Advances in Informatics - 10th Panhellenic Conference on Informatics*, volume 3746 of *Lecture Notes in Computer Science*, chapter Part-of-Speech Tagger for Biomedical Text, Advances in Informatics, pages 382–392. Springer, Berlin/Heidelberg.
- C.J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London.