Learning the Scope of Negation in Biomedical Texts

Roser Morante†, Anthony Liekens‡, Walter Daelemans†
CNTS - Language Technology Group†, Applied Molecular Genomics Group‡
University of Antwerp
Prinsstraat 13, B-2000 Antwerpen, Belgium
{Roser.Morante,Anthony.Liekens,Walter.Daelemans}@ua.ac.be

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 these negation signals. Our approach to negation detection differs in two main aspects from existing research on negation. 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 classifiers, 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.

Szarvas et al. (2008) report that 13.45% of the sentences in the abstracts section of the BioScope corpus and 13.76% of the sentences in the full papers section contain negations. A system that does not deal with negation would treat these cases as false positives.

The goals of this research are to model the scope finding task as a classification task similar to the semantic role labeling task, and to test the performance of a memory–based system that finds the scope of negation signals. Memory-based language processing (Daelemans and van den Bosch, 2005) 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

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 715–724,
Honolulu, October 2008.

argued that lazy learning is at an advantage in solving these highly disjunctive learning problems compared to eager learning, as the latter eliminates 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, from phonology to semantic analysis, such as semantic role labeling (Morante et al., 2008).

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

Negation has been a neglected area in open-domain natural language processing. Most research has been performed in the biomedical domain and has focused 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. The reported results are 95.70 recall and 91.80 precision.

Sanchez-Graillet and Poesio (2007) present an analysis of negated interactions in biological texts and a heuristics-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 dyadic parse tree in two phases: First a preprocessor breaks each sentence into text and operators. Then, a rule based system is used to decide 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 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 it is not explained how the results presented for negation recognition (57% of negations correctly recognised) are evaluated.

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

Golding and Chapman (2003) experiment with 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 identifying 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 determining whether a term is negated or not, we are not aware of research that has focused on learning the full scope of negation signals inside or outside biomedical natural language processing. 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)[1], a freely available resource that consists of medical and biological texts. Every sentence is annotated with information about negation and speculation that indicates the boundaries of the scope and the keywords, as shown in (1).

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

A first characteristic of the annotation of scope in the BioScope corpus is that all sentences that assert the non-existence or uncertainty of something are annotated, in contrast to other corpora where only sentences of interest in the domain are annotated. A second characteristic is that the annotation is extended to the biggest syntactic unit possible so that scopes have the maximal length. In (2) below, negation signal *no* scopes over *primary impairment of glucocorticoid metabolism* instead of scoping only over *primary*.

(2) There is [no] primary impairment of glucocorticoid metabolism in the asthmatics.

The part used in our experiments are the biological paper abstracts from the GENIA corpus (Collier et al., 1999). This part consists of 11,872 sentences in 1,273 abstracts. We automatically discarded five sentences due to annotation errors. The total number of words used is 313,222, 1,739 of which are negation signals that belong to the different types 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[2]. 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. Following the standard format of the CoNLL Shared Task 2006 (Buchholz and Marsi, 2006), 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 (Inside, Outside, Begin) tags produced by the GENIA tagger that indicate if a token is inside a certain chunk, outside, or at the beginning.
8. NE TAG: IOB named entity tags produced by the GENIA tagger that indicate if a token is in-

[1] Web page: `www.inf.u-szeged.hu/rgai/bioscope`.

[2] The accuracy of the tagger might be inflated due to the fact that it was trained on the GENIA corpus.

ABSTR	SNT	TOK	FORM	LEMMA	POS	CHUNK	NE	NEG	NEG SCOPE
ID	ID	ID			TAG	TAG	TAG	SGN	
10415075	07	1	NF-kappa	NF-kappa	NN	B-NP	B-protein	_	I-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 I-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 BioScope corpus converted into columns format.

side a certain named entity, outside, or at the beginning.

9. NEG SIGNAL: tokens that are negation signals are marked as NEG. Negation signals in the BioScope corpus are not always single words, like 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: IO tags that indicate if a token is inside the negation scope (I-NEG), or outside (O-NEG). These tags have been obtained by converting the xml files of BioScope. Each token can have one or more NEG SCOPE tags, depending on the number of 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). This happens as many times as there are negation signals in the sentence. Our conception of the task is inspired by Ramshaw and Marcus' representation of text chunking as a tagging problem (Ramshaw and Marcus, 1995) .

The information that can be used to train the system appears in columns 1 to 8 of Table 1. The information to be predicted by the system is contained in columns 9 and 10.

As far as we know, approaching the negation scope finding task as a token per token classification task is novel, whereas at the same time it conforms to the well established standards of the recent CoNLL Shared Tasks[3] on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007) and semantic role labeling (Surdeanu et al., 2008). 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, and by conforming to existing standards we show that learning the scope of negation can be integrated in a joint learning task with dependency parsing and semantic role labeling.

[3] 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. Additionally, the output of the second classifier is postprocessed with an algorithm that converts non-consecutive blocks of scope into consecutive, as explained in Section 5.3.

As for the first and second phases, we use a memory–based 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). Similarity is defined by computing (weighted) overlap of the feature values of a test instance and training instances. The metric combines a per-feature value distance metric (Cost and Salzberg, 1993) with gain ratio (Quinlan, 1993) based 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 whether a token is a negation signal or not. The memory-based classifier was parameterised by using overlap as the similarity metric, gain ratio for feature weighting, and 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:

- Of the token: Form, lemma, part of speech, and chunk IOB tag.
- Of the token context: Form, POS, and IOB tag of the three previous and three next tokens.

5.2 Scope finding

In the first step of this phase, a classifier predicts whether a token is in the scope of each of the negation signals of a sentence. A pair of a negation signal and a token from the sentence represents an instance. This means that all tokens in a sentence are paired with all negation signals that occur in the sentence. For example, token *NF-kappa* in Table 1 will be represented in two instances as shown in (3). An instance represents the pair [NF–KAPPA, absent] and another one represents the pair [NF–KAPPA, not].

(3) NF-kappa absent [features] I-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 memory–based algorithm was parameterised in this case 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:

- Of the negation signal: Form, POS, chunk IOB tag, type of chunk (NP, VP, ...), and form, POS, chunk IOB tag, type of chunk, and named entity of the 3 previous and 3 next tokens.
- Of the paired token: form, POS, chunk IOB tag, type of chunk, named entity, and form, POS, chunk IOB tag, type of chunk, and named entity type of the 3 previous and 3 next tokens.
- Of the tokens between the negation signal and the token in focus: Chain of POS types, distance in number of tokens, and chain of chunk IOB tags.
- Others: A binary feature indicating whether the token and the negation signal are in the same chunk, and location of the token relative to the negation signal (pre, post, same).

5.3 Post-processing

Negation signals in the BioScope corpus always have one consecutive block of scope tokens, including the signal token itself. However, the scope finding classifier can make predictions that result in non-consecutive blocks of scope tokens: we observed that 54% of scope blocks predicted by the system given gold standard negation signals are non–consecutive. This is why in the second step of the scope finding phase, we apply a post-processing 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 class label for a given negation signal. Post-processing ensures that the resulting scope is one consecutive block of tokens.

In the BioScope corpus negation signals are inside of their scope. The post-processing algorithm that we apply first checks if the negation signal is in its scope. If the signal is out, the algorithm overwrites the predicted scope in order to include the signal in its scope.

Given the position of the signal in the sentence, the algorithm locates the starting and ending tokens of the consecutive block of predicted scope tokens that surrounds the signal. Other blocks of predicted scope tokens may have been predicted outside of this block, but they are separated from the current block, which contains the signal, by tokens that have been predicted not to be in the scope of the negation, as in Figure 1.

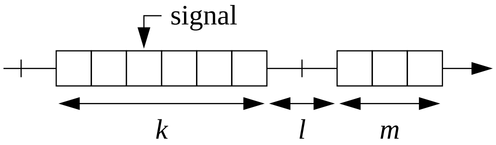

Figure 1: Non-consecutive blocks of scope tokens. For a signal, two blocks of $k = 6$ and $m = 3$ tokens are predicted to be the scope of the signal token, but they are separated by $l = 2$ tokens that are predicted to be out of scope.

The post-processing algorithm decides whether the detached blocks should be connected as one consecutive block of scope tokens, or whether the detached block of scope tokens should be discarded from the scope. Dependent on this decision, either the classification of the separated blocks, or the separating non-scope tokens are considered noisy, and their classification is updated to produce one consecutive block of scope tokens for each signal. This check is performed iteratively for all detached blocks of scope tokens.

As in Figure 1, consider a sentence where the negation signal is in one block K of predicted scope of length k tokens and another block M of m consecutive tokens that is predicted as scope but is separated from the latter scope block by l out-of-scope tokens.

If non-consecutive blocks are near each other, i.e., if l is sufficiently small in comparison with k and m, then the intermediate tokens that have been predicted out of scope could be considered as noise and converted into scope tokens. In contrast, if there are too many intermediate tokens that separate the two blocks of scope tokens, then the additional block of scope is probably wrongly classified.

Following this logic, if $l < \alpha(k + m)$, with a specifically chosen α, the intermediate out-of-scope tokens are re-classified 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-classified 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.

Our implementation first looks for separated blocks from right to left, and then from left to right. 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 from right to left, and $\alpha = 0.3$ for extending the block from left to right into the sentence provide the best results. Algorithm 1 details the above procedure in pseudo code.

Algorithm 1 Post-processing

```
$K \leftarrow$ scope block that contains signal
**while** $M \leftarrow$ nearest separated scope block **do**
    $L \leftarrow$ non-scope block between $K$ and $M$
    **if** $|L| < \alpha(|K| + |M|)$ **then**
        include $L$ in scope
    **else**
        exclude $M$ from scope
    **end if**
    $K \leftarrow$ scope block that contains signal
**end while**
```

6 Results

The results 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, a negation token is correctly classified if it has been assigned a NEG class. In the scope finding task, a token is correctly classified if all the IO 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 IO tag for as many negation signals as there are. For example, token NF-kappa from Table 1 reproduced in (4) will not be correct if it is assigned classes I-NEG I-NEG or O-NEG I-NEG.

(4) 10415075 07 1 NF-kappa NF-kappa NN B-NP B-protein _ I-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, which account for 72.80% of the negation signals. 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 overall results of the negation signal finding system and the results per negation signal. With F1 94.40, it outperforms Baseline 2 by 2.86 points. Precision and recall are very similar. Scores show a clear unbalance between different negation signals. Those with the lowest frequencies get lower scores than those with the highest frequencies. Probably, this could be avoided by training the system with a bigger corpus.

However, a bigger corpus would not help solve all the errors because some of them are caused by inconsistency in the annotation. For example, *absence* is annotated as a negation signal in 57 cases, whereas in 22 cases it is not annotated as such, although in all cases it is used as a negation signal. Example 5 (a) shows one of the 22 cases of *absence* that has not been annotated, and Example 5 (b) shows one of the 57 cases of *absence* annotated as a negation signal. Also *fail* is not annotated as a negation signal in 13 cases where it should.

Neg signals	Total	Prec.	Recall	F1
lack (v)	55	100.00	100.00	100.00
neither (con)	34	100.00	100.00	100.00
lack (n)	33	100.00	100.00	100.00
unable	30	100.00	100.00	100.00
neither (det)	8	100.00	100.00	100.00
no (adv)	5	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 (det)	204	95.63	96.56	96.09
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
could	6	66.66	33.33	44.44
absent	13	42.85	23.07	30.00
with	6	0.00	0.00	0.00
either	2	0.00	0.00	0.00
instead	2	0.00	0.00	0.00
never	2	0.00	0.00	0.00
impossible	1	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
or	1	0.00	0.00	0.00
Overall	1739	94.21	94.59	94.40

Table 3: F scores of the negation finding classifier.

(5) (a) Retroviral induction of TIMP-1 not only resulted in cell survival but also in continued DNA synthesis for up to 5 d in the **absence** of serum, while controls underwent apoptosis.

(b) A significant proportion of transcripts appear to terminate prematurely in the <xcope id= X654.8.1 ><cue type= negation ref= X654.8.1 > **absence** </cue> of transactivators </xcope>.

Other negation signals are arbitrarily annotated. *Failure* is annotated as a negation signal in 8 cases where it is followed by a preposition, like in Example 6 (a), and it is not annotated as such in 26 cases, like Example 6 (b), where it is modified by an adjective.

(6) (a) ... the <xcope id= X970.8.2> <cue type= negation ref= X970.8.2>**failure**</cue> of eTh1 cells to produce IL-4 in response to an antigen </xcope> is due, at least partially, to a <xcope id= X970.8.1> < cue type= negation ref= X970.8.1> **failure**</cue> to induce high-level transcription of the IL-4 gene by NFAT </xcope></xcope>.

(b) Positive-pressure mechanical ventilation supports gas exchange in patients with respiratory **failure** but is also responsible for significant lung injury.

The errors in detecting *with* as a negation signal are caused by the fact that it is embedded in the expression *with the exception of*, which occurs 6 times in contrast with the 5265 occurrences of *with*. *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*.

6.2 Scope finding

We provide the results of the classifier and the results of applying the postprocessing 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.

	Prec.	Recall	F1	PCS
GS NEG	86.03	85.53	85.78	39.39
PR NEG	79.83	77.42	78.60	36.31

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 7.18 points lower than the F1 of GS NEG, which is an expected effect due to the performance of classifier that finds negation signals. 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 postprocessing algorithm. The most remarkable result is the 29.60 and 21.58 error reduction in the percentage of fully correct scopes of GS NEG and PR NEG respectively, which shows that the algorithm is efficient. Also interesting is the increase in F1 of GS NEG and PR NEG.

	Prec.	Recall	F1	PCS
GS NEG	88.63	88.17	88.40	57.33
PR NEG	80.70	81.29	80.99	50.05

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

Table 6 shows detailed results of the system based on predicted negation signals after applying the postprocessing algorithm. Classes O-NEG and I-NEG are among the most frequent and get high scores. Classes composed only of O-NEG tags are easier to predict.

Scope tags	Total	Prec.	Recall	F1
O-NEG	29590	86.78	84.75	85.75
O-NEG O-NEG O-NEG	46	100.00	63.04	77.33
I-NEG	12990	73.41	80.72	76.89
O-NEG O-NEG	2848	84.11	68.43	75.46
I-NEG I-NEG O-NEG	69	62.92	81.15	70.88
I-NEG I-NEG	684	57.30	65.93	61.31
I-NEG O-NEG O-NEG	20	50.00	75.00	60.00
O-NEG I-NEG	791	72.13	50.06	59.10
I-NEG O-NEG	992	45.32	67.94	54.37
O-NEG I-NEG I-NEG	39	100.00	20.51	34.04
I-NEG I-NEG I-NEG	22	26.66	36.36	30.76
O-NEG O-NEG I-NEG	14	0.00	0.00	0.00
Overall	48105	80.70	81.29	80.99

Table 6: F scores of the system per scope class after applying the postprocessing algorithm.

Table 7 shows information about the percentage of correct scopes per negation signal after applying the algorithm to PR-NEG. A clear example of an incorrect prediction is the occurrence of *box* in the list. The signal with the highest percentage of PCS is *without*, followed by *no* (determiner), *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.

Neg signals	Total	Correct	PCS
without	82	56	68.29
no (det)	206	133	64.56
rather	20	11	55.00
not	1066	556	52.15
neither (det)	8	4	50.00
none	7	3	42.85
neither (conj)	34	16	47.05
no (adv)	5	2	40.00
fail	63	23	36.50
missing	3	1	33.33
absence	68	22	32.35
lack (v.)	54	17	31.48
absent	7	2	28.57
lack (n.)	33	9	27.27
nor	43	11	25.58
unable	30	8	26.66
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	874	50.05

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

7 Conclusions

Given the fact that a significant portion of biomedical text is negated, recognising negated instances is important in NLP applications. 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 classifiers, 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 classifier achieves 94.40 F1, and the second 80.99. However, the evaluation in terms of correct scopes shows the weakness of the system. This is why a postprocessing algorithm is applied. The algorithm achieves an error reduction of 21.58, with 50.05 % of fully correct scopes in the system based on predicted negation signals.

These results suggest that unsupervised machine learning algorithms are suited for tackling the task, as it was expected from results obtained in other natural language processing tasks. However, results also suggest that there is room for improvement. A first improvement would consist in predicting the scope chunk per chunk instead of token per token, because most negation scope boundaries coincide with boundaries of chunks.

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. Unfortunately, there are no previous comparable approaches to measure the quality of our results.

Additionally, we have shown that negation finding can be modelled as a classification task in a way similar to other linguistic tasks like semantic role labeling. In our model, tokens of a sentence are classified as being a negation signal or not, and as being inside or outside the scope of the negation signal(s). This representation would allow to integrate the task with other semantic tasks and exploring the interaction between different types of knowledge in a joint learning setting.

Further research is possible in several directions. In the first place, other machine learning algorithms could be integrated in the system in order to optimise performance. Secondly, the system should be tested in different types of biomedical texts, like full papers or medical reports to check its robustness. Finally, the postprocessing algorithm could be improved by using more sophisticated sequence classification techniques (Dietterich, 2002) .

Acknowledgments

Our work was made possible through financial support from the University of Antwerp (GOA project BIOGRAPH). We are thankful to three anonymous reviewers for their valuable comments and suggestions.

References

M. Averbuch, T. Karson, B. Ben-Ami, O. Maimon, and L. Rokach. 2004. Context-sensitive medical information retrieval. In *Proc. 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 *Proc. of International Workshop Language and Speech Infrastructure for Information Access in the Balkan Countries*, pages 1–8, Borovets, Bulgaria.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of the X CoNLL Shared Task*, New York. SIGNLL.

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*.

S. Cost and S. Salzberg. 1993. A weighted nearest neighbour algorithm for learning with symbolic features. *Machine learning*, 10:57–78.

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.

T. G. Dietterich. 2002. Machine learning for sequential data: A review. In *Lecture Notes in Computer Science 2396*, pages 15–30, London. Springer Verlag.

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).

l. 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.

R. Morante, W. Daelemans, and V. Van Asch. 2008. A combined memory-based semantic role labeler of english. In *Proc. of the CoNLL 2008*, pages 208–212, Manchester, UK.

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.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL-2007 shared task on dependency parsing. In *Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague.

J.R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

L. Ramshaw and M. Marcus. 1995. Text chunking using transformation-based learning. In *Proc. of ACL Third Workshop on Very Large Corpora*, pages 82–94, Cambridge, MA. ACL.

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.

M. Surdeanu, R. Johansson, A. Meyers, Ll. Màrquez, and J. Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proc. of CoNLL 2008*, pages 159–177, Manchester, UK.

G. Szarvas, V. Vincze, R. Farkas, and J. Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scopein biomedical texts. In *Proc. of BioNLP 2008*, 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 *Proc. 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.