Exploiting Multi-Features to Detect Hedges and Their Scope in Biomedical Texts

Huiwei Zhou[1], Xiaoyan Li[2], Degen Huang[3], Zezhong Li[4], Yuansheng Yang[5]
Dalian University of Technology
Dalian, Liaoning, China
{[1]zhouhuiwei, [3]huangdg, [5]yangys}@dlut.edu.cn
[2]lixiaoyan@mail.dlut.edu.cn
[4]lizezhonglaile@163.com

Abstract

In this paper, we present a machine learning approach that detects hedge cues and their scope in biomedical texts. Identifying hedged information in texts is a kind of semantic filtering of texts and it is important since it could extract speculative information from factual information. In order to deal with the semantic analysis problem, various evidential features are proposed and integrated through a Conditional Random Fields (CRFs) model. Hedge cues that appear in the training dataset are regarded as keywords and employed as an important feature in hedge cue identification system. For the scope finding, we construct a CRF-based system and a syntactic pattern-based system, and compare their performances. Experiments using test data from CoNLL-2010 shared task show that our proposed method is robust. F-score of the biological hedge detection task and scope finding task achieves 86.32% and 54.18% in in-domain evaluations respectively.

1. Introduction

Identifying sentences in natural language texts which contain unreliable or uncertain information is an increasingly important task of information extraction since the extracted information that falls in the scope of hedge cues cannot be presented as factual information. Szarvas et al. (2008) report that 17.69% of the sentences in the abstracts section of the BioScope corpus and 22.29% of the sentences in the full papers section contain hedge cues. Light et al. (2004) estimate that 11% of sentences in MEDLINE abstracts contain speculative fragments. Szarvas (2008) reports that 32.41% of gene names mentioned in the hedge classification dataset described in Medlock and Briscoe (2007) appear in a speculative sentence. Many Wikipedia articles contain a specific *weasel tag* which mark sentences as non-factual (Ganter and Strube, 2009).

There are some Natural Language Processing (NLP) researches that demonstrate the benefit of hedge detection experimentally in several subjects, such as the ICD-9-CM coding of radiology reports and gene named Entity Extraction (Szarvas, 2008), question answering systems (Riloff et al., 2003), information extraction from biomedical texts (Medlock and Briscoe, 2007).

The CoNLL-2010 Shared Task (Farkas et al., 2010) “Learning to detect hedges and their scope in natural language text” proposed two tasks related to speculation research. Task 1 aimed to identify sentences containing uncertainty and Task 2 aimed to resolve the in-sentence scope of hedge cues. We participated in both tasks.

In this paper, a machine learning system is constructed to detect sentences in texts which contain uncertain or unreliable information and to find the scope of hedge cues. The system works in two phases: in the first phase uncertain sentences are detected, and in the second phase in-sentence scopes of hedge cues are found. In the uncertain information detecting phase, hedge cues play an important role. The sentences that contain at least one hedge cue are considered as uncertain, while sentences without cues are considered as factual. Therefore, the task of uncertain information detection can be converted into the task of hedge cue identification. Hedge cues that appear in the training dataset are collected and used as keywords to find hedges. Furthermore, the detected keywords are employed as an important feature in hedge cue identification system. In addition to keywords, various evidential features are proposed and integrated through a machine learning model. Finding the scope of a hedge cue is to determine at sentence level which words are affected by the

Proceedings of the Fourteenth Conference on Computational Natural Language Learning: Shared Task, pages 106–113,
Uppsala, Sweden, 15-16 July 2010.

hedge cue. In the scope finding phase, we construct a machine learning-based system and a syntactic pattern-based system, and compare their performances.

For the learning algorithm, Conditional random fields (CRFs) is adopted relying on its flexible feature designs and good performance in sequence labeling problems as described in Lafferty et al. (2001). The main idea of CRFs is to estimate a conditional probability distribution over label sequences, rather than over local directed label sequences as with Hidden Markov Models (Baum and Petrie, 1966) and Maximum Entropy Markov Models (McCallum et al., 2000).

Evaluation is carried out on the CoNLL-2010 shared task (Farkas et al., 2010) dataset in which sentences containing uncertain information are annotated. For the task of detecting uncertain information, uncertain cues are annotated. And for the task of finding scopes of hedge cues, hedge cues and their scope are annotated as shown in sentence (a): hedge cue *indicate that*, and its scope *indicate that dhtt is widely expressed at low levels during all stages of Drosophila development* are annotated.

(a)*Together, these data <xcope id="X8.74.1"><cue ref="X8.74.1" type="speculation">indicate that</cue> dhtt is widely expressed at low levels during all stages of Drosophila development</xcope>.*

2. Related Work

In the past few years, a number of studies on hedge detection from NLP perspective have been proposed. Elkin et al. (2005) exploited handcrafted rule-based negation/uncertainty detection modules to detect the negation or uncertainty information. However, their detection modules were hard to develop due to the lack of standard corpora that used for evaluating the automatic detection and scope resolution. Szarvas et al. (2008) constructed a corpus annotated for negations, speculations and their linguistic scopes. It provides a common resource for the training, testing and comparison of biomedical NLP systems.

Medlock and Briscoe (2007) proposed an automatic classification of hedging in biomedical texts using weakly supervised machine learning. They started with a very limited amount of annotator-labeled seed data. Then they iterated and acquired more training seeds without much manual intervention. The best classifier using their model achieved 0.76 precision/recall break-even-point (BEP). Further, Medlock (2008) illuminated the hedge identification task including annotation guidelines, theoretical analysis and discussion. He argued for separation of the acquisition and classification phases in semi-supervised machine learning method and presented a probabilistic acquisition model. In probabilistic model he assumed bigrams and single terms as features based on the intuition that many hedge cues are bigrams and single terms and achieves a peak performance of around 0.82 BEP.

Morante and Daelemans (2009) presented a meta-learning system that finds the scope of hedge cues in biomedical texts. The system worked in two phases: in the first phase hedge cues are identified, and in the second phase the full scopes of these hedge cues are found. The performance of the system is tested on three subcorpora of the BioScope corpus. In the hedge finding phase, the system achieves an F-score of 84.77% in the abstracts subcorpus. In the scope finding phase, the system with predicted hedge cues achieves an F-score of 78.54% in the abstracts subcorpus.

The research on detecting uncertain information is not restricted to analyze biomedical documents. Ganter and Strube (2009) investigated Wikipedia as a source of training data for the automatic hedge detection using word frequency measures and syntactic patterns. They showed that the syntactic patterns worked better when using the manually annotated test data, word frequency and distance to the weasel tag was sufficient when using Wikipedia weasel tags themselves.

3. Identifying Hedge Cues

Previous studies (Light et al., 2004) showed that the detection of hedging could be solved effectively by looking for specific keywords which were useful for deciding whether a sentence was speculative. Szarvas (2008) reduces the number of keyword candidates without excluding helpful keywords for hedge classification. Here we also use a simple keyword-based hedge cue detection method.

3.1 Keyword-based Hedge Cue Detection

In order to recall as many hedge cues as possible, all hedge cues that appear in the training dataset are used as keywords. Hedge cues are represented by one or more tokens. The list of all hedge cues in the training dataset is comprised of 143 cues. 90 hedge cues are unigrams, 24 hedge cues are bigrams, and the others are trigrams, four-grams and five-grams. Besides, hedge cues that appear in the training dataset and their synonyms in WordNet[1] are also selected as keywords for hedge cue detection. The complete list of them contains 438 keywords, 359 of which are unigrams. Many tokens appear in different grams cues, such as *possibility* appears in five-grams cue *cannot rule out the possibility*, four-gram cue *cannot exclude the possibility*, trigrams cue *raise the possibility* and unigram cue *possibility*. To find the complete cues, keywords are matched through a maximum matching method (MM) (Liu et al., 1994). For example, though *indicate* and *indicate that* are both in keywords list, *indicate that* is extracted as a keyword in sentence (a) through MM.

3.2 CRF-based Hedge Cue Detection

Candidate cues are extracted based on keywords list in keyword-based hedge cue detection stage. But the hedge cue is extremely ambiguous, so CRFs are applied to correct the false identification results that occurred in the keyword-based hedge cue detection stage. The extracted hedge cues are used as one feature for CRFs-based hedge cue detection.

A CRF identifying model is generated by applying a CRF tool to hedge cue labeled sequences. Firstly, hedge cue labeled sentences are transformed into a set of tokenized word sequences with IOB2 labels:

B-cue Current token is the beginning of a hedge cue

I-cue Current token is inside of a hedge cue

O Current token is outside of any hedge cue

For sentence (a) the system assigns the B-cue tag to *indicate*, the I-cue tag to *that* and the O tag to the rest of tokens as shown in Figure1.

The hedge cues that are found by keyword-based method is also given IOB2 labels feature as shown in Figure1.

Text	**Keyword Labels Feature**	**Cue Labels**
…	...	...
these	O	O
data	O	O
indicate	B	B-cue
that	I	I-cue
dhtt	O	O
is	O	O
...	...	...

Figure 1: Example of Cues labels and Keywords labels Feature

Diverse features including keyword feature are employed to our CRF-based hedge cue detection system.

(1) Word Features

• *Word (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

Where *Word (0)* is the current word, *Word (-1)* is the first word to the left, *Word (1)* is the first word to the right, etc.

(2) Stem Features

The motivation for stemming in hedge identification is that distinct morphological forms of hedge cues are used to convey the same semantics (Medlock, 2008). In our method, GENIA Tagger[2] (Tsuruoka et al., 2005) is applied to get stem features.

• *Stem (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

where *Stem (0)* is the stem for the current word, *Stem(-1)* is the first stem to the left, *Stem (1)* is the first stem to the right, etc.

(3) Part-Of-Speech Features

Since most of hedge cues in the training dataset are verbs, auxiliaries, adjectives and adverbs. Therefore, Part-of-Speech (POS) may provide useful evidence about the hedge cues and their boundaries. GENIA Tagger is also used to generate this feature.

• *POS (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

where *POS (0)* is the current POS, *POS (-1)* is the first POS to the left, *POS (1)* is the first POS to the right, etc.

(4) Chunk Features

Some hedge cues are chunks consisting of more than one token. Chunk features may contribute to the hedge cue boundaries. We use GENIA Tagger to get chunk features for each token. The

[1] Available at http://wordnet.princeton.edu/

[2] Available at http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

chunk features include unigram, bigram, and trigram types, listed as follows:

- *Chunk (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*
- *Chunk (i−1)+Chunk(i) (i =−1,0,+1,+2)*
- *Chunk (i−2) + Chunk (i−1)+Chunk (i) (i= 0,+1,+2)*

where *Chunk (0)* is the chunk label for the current word, *Chunk (−1)* is the chunk label for the first word to the left , *Chunk (1)* is the chunk label for the first word to the right, etc.

(5) Keyword Features

Keyword labels feature is an important feature.

- *Keyword (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

where *Keyword (0)* is the current keyword label, *Keywords (-1)* is the keyword label for the first keyword to the left, *Keywords (1)* is the keyword label for the first keyword to the right, etc.

Feature sets can be easily redefined by changing the window size *n*. The relationship of the window size and the F-score observed in our experiments will be reported in Section 5.

4. Hedge Scope Finding

In this task, a CRFs classifier is applied to predict for all the tokens in the sentence whether a token is the first token of the scope sequence (F-scope), the last token of the scope sequence (L-scope), or neither (None). For sentence (a) in Section 1, the classifier assigns F-scope to *indicate*, L-scope to *benchmarks*, and None to the rest of the tokens. Only sentences that assigned cues in the first phase are selected for hedge scope finding. Besides, a syntactic pattern-based system is constructed, and compared with the CRF-based system.

4.1 CRF-based System

The features that used in CRF-based hedge cue detection systems are also used for scope finding except for the keyword features. The features are:

(1) Word Features

- *Word (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

(2) Stem Features

- *Stem (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

(3) Part-Of-Speech Features

- *POS (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

(4) Chunk Features

The chunk features include unigram, bigram, and trigram types, listed as follows:

- *Chunk (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*
- *Chunk (i−1)+Chunk(i) (i =−1,0,+1,+2)*
- *Chunk (i−2) + Chunk (i−1)+Chunk (i) (i= 0,+1,+2)*

(5) Hedge cues Features

Hedge cues labels that are doped out in Task 1 are selected as an important feature.

- *Hedge cues (i) (i=-n, …, −2, −1, 0, +1, +2, …, +n)*

where *Hedge cues (0)* is the cue label for the current word, *Hedge cues (−1)* is the cue label for the first word to the left , *Hedge cues (1)* is the cue label for the first word to the right, etc.

The scope of the sequence must be consistent with the hedge cues. That means that the number of the F-scope and L-scope must be the same with the hedge cues. However, sometimes their number predicted by classifier is not same. Therefore, we need to process the output of the classifier to get the complete sequence of the scope. The following post processing rules are adapted.

- If the number of F-scope, L-scope and hedge cue is the same, the sequence will start at the token predicted as F-scope, and end at the token predicted as L-scope.
- If one token has been predicted as F-scope and none has been predicted as L-scope, the sequence will start at the token predicted as F-scope and end at the end of the sentence. Since when marking the scopes of keywords, linguists always extend the scope to the biggest syntactic unit possible.
- If one token has been predicted as L-scope and none has been predicted as F-scope, the sequence will start at the hedge cue and end at the token predicted as L-scope. Since scopes must contain their cues.
- If one token has been predicted as F-scope and more than one has been predicted as L-scope, the sequence will end at the first token predicted as L-scope. Statistics from prediction on CoNLL-2010 Shared Task evaluation data show that 20 sentences are in this case. And the scope of 6 sentences extends to the first L-scope, and the scope of 3 sentences end at the last L-scope, the others are predicted mistakenly. Our system prediction and gold-standard annotation are shown in sentence (b1) and (b2) respectively.

(b1) our system annotation:
dRas85DV12 <xcope id="X3.64.1"><cue ref="X3.64.1" type="speculation">may</cue> be more potent than dEGFRλ</xcope> because dRas85DV12 can activate endogenous PI3K signaling</xcope> [16].

(b2) gold-standard annotation:
dRas85DV12 <xcope id="X3.64.1"><cue ref="X3.64.1" type="speculation">may</cue> be more potent than dEGFRλ</xcope> because dRas85DV12 can activate endogenous PI3K signaling [16].

- If one token has been predicted as L-scope and more than one has been predicted as F-scope, the sequence will start at the first token predicted as F-scope.
- If an L-scope is predicted before an F-scope, the sequence will start at the token predicted as F-scope, and finished at the end of the sentence.

4.2 Syntactic Pattern-based System

Hedge scopes usually can be determined on the basis of syntactic patterns dependent on the cue. Therefore, a syntactic pattern-based system is also implemented for hedge scope finding. When the sentence is predicted as uncertain, the toolkit of Stanford Parser[3] (Klein and Manning, 2003) is utilized to parse the sentence into a syntactic tree, which can release a lot of information about the grammatical structure of sentences that is beneficial for the finding of hedge scope. For sentence (c) the Stanford Parser gives the syntactic tree as showed in Figure 2.

(c) *This <xcope id="X*.*.*"><cue ref="X*.*.*" type="speculation"> may </cue> represent a viral illness</xcope>*.

It is obvious to see from the syntactic tree, all the words of the parsed sentence concentrate at the places of leaves. We use the following rules to find the scope.

- If the tag of the word is “B-cue”, it is predicted as F-scope.
- If the POS of the hedge cue is verbs and auxiliaries, the L-scope is signed at the end of the clause.
- If the POS of the hedge cue is attributive adjectives, the L-scope is signed at the following noun phrase.
- If the POS of the hedge cue is prepositions, the L-scope is signed at the following noun phrase.
- If none of the above rules apply, the scope of a hedge cue starts with the hedge cue and ends at the following clause.

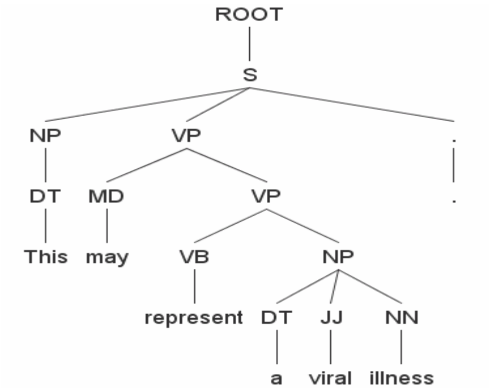

Figure 2: Syntactic tree parsed by Stanford Parser

5. Experiments and Discussion

We evaluate our method using CoNLL-2010 shared task dataset. The evaluation of uncertain information detection task is carried out using the sentence-level F-score of the uncertainty class. As mentioned in Section 1, Task 1 is converted into the task of hedge cues identification. Sentences can be classified as certain or uncertain according to the presence or absence of a few hedge cues within the sentences. In task of finding in-sentence scopes of hedge cues, a scope is correct if all the tokens in the sentence have been assigned the correct scope class for a specific hedge signal.

5.1 Detecting Uncertain Information

In the CoNLL-2010 Shared Task 1, our in-domain system obtained the F-score of 85.77%. Sentence-level results of in-domain systems under the condition *n=3* (window size) are summarized in Table 1.

System	Prec.	Recall	F-score
Keyword-based	41.15	99.24	58.18
CRF-based system (without keyword features)	88.66	80.13	84.18
CRF-based system + keyword features	86.21	84.68	85.44
CRF-based system	86.49	85.06	85.77

[3] Available at http://nlp.stanford.edu/software/lex-parser.shtml

+ keyword features + MM			

Table 1: Official in-domain results for Task 1 (*n=3*)

The keyword-based system extracts hedge cues through maximum matching method (MM). As can be seen in Table 1, the system achieves a high recall (99.24%). This can be explained that almost all of the hedge cues in the test dataset are in the keywords list. However, it also brings about the low precision since not all potential speculative keywords convey real speculation. So the keyword-based method can be combined with our CRF-based method to get better performance.

All the CRF-based systems in Table 1 significantly outperform the keyword-based system, since the multi-features achieve a high precision. And the result with keyword features is better than the result without it. The keyword features improve the performance by recalling 39 true positives. In addition, further improvement is achieved by using Maximum Matching method (MM).

In the test dataset, there should be a few hedge cues not in the training dataset. And the additional resources besides the manually labeled data are allowed for in-domain predictions. Therefore, the synonyms of the keywords can be used for in-domain systems. The synonyms of the keywords are added to the keywords list, and are expected to improve detecting performance. The synonyms are obtained from WordNet.

Table 2 shows the relationship between the window size and the sentence-level results. This table shows the results with and without synonyms. Generally, the results with synonyms are better than the results without them. With respect to window size, the wider the window size, the better precision can be achieved. However, large window size leads to low recall which is probably because of data sparse. The best F-score 86.32 is obtained when the window size is +/-4.

Window size	**Synonyms**	**Prec.**	**Recall**	**F-score**
1	without synonyms	85.27	86.46	85.86
	with synonyms	85.66	86.20	85.93
2	without synonyms	86.35	85.70	86.02
	with	86.14	84.94	85.53
3	without synonyms	86.49	85.06	85.77
	with synonyms	86.69	84.94	85.81
4	without synonyms	86.34	84.81	85.57
	with synonyms	87.21	85.44	**86.32**

Table 2: Sentence-level results relative to synonyms and window size for speculation detection

5.2 Finding Hedge Scope

In the CoNLL-2010 Shared Task 2, our in-domain system obtained the F-score of 44.42%. Table 3 shows the scope finding results. For in-domain scope finding system, we use the hedge cues extracted by the submitted CRF-based in-domain system (the best result 85.77 in Table 1). The result of the syntactic pattern-based system is not ideal probably due to the syntactic parsing errors and limited annotation rules.

System	**Prec.**	**Recall**	**F-score**
syntactic pattern-based	44.31	42.59	43.45
CRF-based	45.32	43.56	44.42

Table 3: Official in-domain results for Task 2

Through analyzing the false of our scope finding system, we found that many of our false scope were caused by such scope as sentence (d1) shows. Our CRF-based system signed the L-scope to the end of sentence mistakenly. The incorrectly annotation of our system and gold-standard annotation are shown in sentence (d1) and (d2) respectively. So an additional rule is added to our CRF-based system to correct the L-scope. The rule is:

- If one token has been predicted as L-scope, and if the previous token is “)”, or “]”, the L-scope will be modified just before the paired token “(” or “[”.

(d1) The incorrectly predicted version:
These factors were <cue ref="X1.178.1" type="speculation">presumed</cue> to be pathogenic</xcope> (85).
(d2) Gold-standard annotation:
These factors were <cue ref="X1.178.1" type="speculation">presumed</cue> to be pathogenic (85) </xcope>.

F-score is reached to 51.83 by combining this additional rule with the submitted CRF-based in-domain system as shown in Table 4.

TP	FP	FN	Prec.	Recall	F-score
525	468	508	52.87	50.82	51.83

Table 4: Official in-domain results for Task 2

Several best results of Task 1 are exploited to investigate the relationship between the window size and the scope finding results. From the results of Table 5, we can see that the case of *n=4* gives the best precision, recall and F-score. And the case of *n=2* and the case of *n=3* based on the same task 1 system have a very similar score. With respect to the different systems of Task 1, in principle, the higher the F-score of Task 1, the better the performance of Task 2 can be expected. However, the result is somewhat different from the expectation. The best F-score of Task 2 is obtained under the case *F-score (task 1) =86.02*. This indicates that it is not certain that Task 2 system based on the best Task 1 result gives the best scope finding performance.

F-score (Task 1)	Window size	Prec.	Recall	F-score
86.32	4 3 2	54.32 52.59 52.90	51.69 50.05 50.34	52.98 51.29 51.59
86.02	4 3 2	54.85 53.13 53.13	52.57 50.92 50.92	**53.68** 52.00 52.00
85.86	4 3 2	54.19 52.50 52.50	52.57 50.92 50.92	53.37 51.70 51.70

Table 5: Scope finding results relative to the results of task 1 and window size

In the case that scopes longer than *n* (window size) words, the relevant cue will thus not fall into the +/-n word window of the L-scope and all hedge cue features will be O tag. The hedge cue features will be useless for detecting L-scopes. Taking into account the importance of hedge cue features, the following additional features are also incorporated to capture hedge cue features.

- *Distance to the closest preceding hedge cue*
- *Distance to the closest following hedge cue*
- *Stem of the closest preceding hedge cue*
- *Stem of the closest following hedge cue*
- *POS of the closest preceding hedge cue*
- *POS of the closest following hedge cue*

Table 6 shows the results when the additional hedge cue features are used. The results with additional hedge cue feature set are constantly better than the results without them. In most of cases, the improvement is significant. The best F-score 54.18% is achieved under the case *F-score (task 1) =86.02* and *n=4.*

F-score (Task 1)	Window size	Prec.	Recall	F-score
86.32	4 3 2	54.73 54.22 53.41	52.08 51.60 50.82	53.37 52.88 52.08
86.02	4 3 2	55.35 54.75 53.94	53.05 52.47 51.69	**54.18** 53.58 52.79
85.86	4 3 2	54.49 53.79 53.09	52.86 52.18 51.50	53.66 52.97 52.29

Table 6: Scope finding results relative to the results of Task 1 and window size with additional cue features

The upper-bound results of CRF-based system assuming gold-standard annotation of hedge cues are show in Table 7.

TP	FP	FN	Prec.	Recall	F-score
618	427	415	59.14	59.83	59.48

Table 7: Scope finding result with gold-standard hedge signals

A comparative character analysis of syntactic pattern-based method and CRF-based method will be interesting, which can provide insights leading to better methods in the future.

6. Conclusion

In this paper, we have exploited various useful features evident to detect hedge cues and their scope in biomedical texts. For hedge detection task, keyword-based system is integrated with CRF-based system by introducing keyword features to CRF-based system. Our experimental results show that the proposed method improves the performance of CRF-based system by the additional keyword features. Our system has achieved a state of the art F-score 86.32% on the sentence-level evaluation. For scope finding task, two different systems are established: CRF-based and syntactic pattern-based system. CRF-based system outperforms syntactic pattern-based system due to its evidential features.

In the near future, we will improve the hedge cue detection performance by investigating more implicit information of potential keywords. On the other hand, we will study on how to improve scope finding performance by integrating CRF-based and syntactic pattern-based scope finding systems.

References

Leonard E. Baum, and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563.

Peter L. Elkin, Steven H. Brown, Brent A. Bauer, Casey S. Husser, William Carruth, Larry R. Bergstrom, and Dietlind L. Wahner-Roedler. 2005. A controlled trial of automated classification of negation from clinical notes. *BMC Medical Informatics and Decision Making*, 5(13).

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. 2010. The CoNLL-2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In *Proceedings of CoNLL-2010: Shared Task, 2010*, pages 1–12.

Viola Ganter, and Michael Strube. 2009. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 173–176.

Dan Klein, and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Marc Light, Xin Ying Qiu, and Padmini Srinivasan. 2004. The language of bioscience: facts, speculations, and statements in between. In *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 17–24.

Yuan Liu, Qiang Tan, and Kunxu Shen. 1994. *The word segmentation rules and automatic word segmentation methods for Chinese information processing*. QingHua University Press and GuangXi Science and Technology Press.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of ICML 2000*, pages 591–598.

Ben Medlock. 2008. Exploring hedge identification in biomedical literature. *Journal of Biomedical Informatics*, 41(4):636–654.

Ben Medlock, and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of ACL-07*, pages 992–999.

Roser Morante, and Walter Daelemans. 2009. Learning the scope of hedge cues in biomedical texts. In *Proceedings of the Workshop on BioNLP, ACL 2009*, pages 28–36.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 25–32.

György Szarvas. 2008. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of ACL: HLT*, pages 281–289.

György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. In *Proceedings of BioNLP 2008: Current Trends in Biomedical Natural Language*, pages 38–45.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, Jun´ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Advances in Informatics 2005*, pages 382–392.