Hedge classification in biomedical texts with a weakly supervised selection of keywords

György Szarvas
Research Group on Artificial Intelligence
Hungarian Academy of Sciences / University of Szeged
HU-6720 Szeged, Hungary
szarvas@inf.u-szeged.hu

Abstract

Since facts or statements in a hedge or negated context typically appear as false positives, the proper handling of these language phenomena is of great importance in biomedical text mining. In this paper we demonstrate the importance of hedge classification experimentally in two real life scenarios, namely the ICD-9-CM coding of radiology reports and gene name Entity Extraction from scientific texts. We analysed the major differences of speculative language in these tasks and developed a maxent-based solution for both the free text and scientific text processing tasks. Based on our results, we draw conclusions on the possible ways of tackling speculative language in biomedical texts.

1 Introduction

The highly accurate identification of several regularly occurring language phenomena like the speculative use of language, negation and past tense (temporal resolution) is a prerequisite for the efficient processing of biomedical texts. In various natural language processing tasks, relevant statements appearing in a speculative context are treated as false positives. Hedge detection seeks to perform a kind of semantic filtering of texts, that is it tries to separate factual statements from speculative/uncertain ones.

1.1 Hedging in biomedical NLP

To demonstrate the detrimental effects of speculative language on biomedical NLP tasks, we will consider two inherently different sample tasks, namely the ICD-9-CM coding of radiology records and gene information extraction from biomedical scientific texts. The general features of texts used in these tasks differ significantly from each other, but both tasks require the exclusion of uncertain (or speculative) items from processing.

1.1.1 Gene Name and interaction extraction from scientific texts

The test set of the hedge classification dataset [1] (Medlock and Briscoe, 2007) has also been annotated for gene names[2].

Examples of speculative assertions:
Thus, the D-mib wing phenotype may result from defective N inductive signaling at the D-V boundary.
A similar role of Croquemort has not yet been tested, but seems likely since the crq mutant used in this study (crqKG01679) is lethal in pupae.

After an automatic parallelisation of the 2 annotations (sentence matching) we found that a significant part of the gene names mentioned (638 occurences out of a total of 1968) appears in a speculative sentence. This means that approximately 1 in every 3 genes should be excluded from the interaction detection process. These results suggest that a major portion of system false positives could be due to hedging if hedge detection had been neglected by a gene interaction extraction system.

1.1.2 ICD-9-CM coding of radiology records

Automating the assignment of ICD-9-CM codes for radiology records was the subject of a shared task

[1] http://www.cl.cam.ac.uk/~bwm23/
[2] http://www.cl.cam.ac.uk/~nk304/

Proceedings of ACL-08: HLT, pages 281–289,
Columbus, Ohio, USA, June 2008.

challenge organised in Spring 2007. The detailed description of the task, and the challenge itself can be found in (Pestian et al., 2007) and online[3]. ICD-9-CM codes that are assigned to each report after the patient's clinical treatment are used for the reimbursement process by insurance companies. There are official guidelines for coding radiology reports (Moisio, 2006). These guidelines strictly state that an uncertain diagnosis should never be coded, hence identifying reports with a diagnosis in a speculative context is an inevitable step in the development of automated ICD-9-CM coding systems. The following examples illustrate a typical non-speculative context where a given code should be added, and a speculative context where the same code should never be assigned to the report:

non-speculative: *Subsegmental **atelectasis** in the left lower lobe, otherwise normal exam.*

speculative: *Findings suggesting viral or reactive airway disease with right lower lobe **atelectasis** or pneumonia.* In an ICD-9 coding system developed for the challenge, the inclusion of a hedge classifier module (a simple keyword-based lookup method with 38 keywords) improved the overall system performance from 79.7% to 89.3%.

1.2 Related work

Although a fair amount of literature on hedging in scientific texts has been produced since the 1990s (e.g. (Hyland, 1994)), speculative language from a Natural Language Processing perspective has only been studied in the past few years. This phenomenon, together with others used to express forms of authorial opinion, is often classified under the notion of subjectivity (Wiebe et al., 2004), (Shanahan et al., 2005). Previous studies (Light et al., 2004) showed that the detection of hedging can be solved effectively by looking for specific keywords which imply that the content of a sentence is speculative and constructing simple expert rules that describe the circumstances of where and how a keyword should appear. Another possibility is to treat the problem as a classification task and train a statistical model to discriminate speculative and non-speculative assertions. This approach requires the availability of labeled instances to train the models on. Riloff et al. (Riloff et al., 2003) applied bootstrapping to recognise subjective noun keywords and classify sentences as subjective or objective in newswire texts. Medlock and Briscoe (Medlock and Briscoe, 2007) proposed a weakly supervised setting for hedge classification in scientific texts where the aim is to minimise human supervision needed to obtain an adequate amount of training data.

Here we follow (Medlock and Briscoe, 2007) and treat the identification of speculative language as the classification of sentences for either speculative or non-speculative assertions, and extend their methodology in several ways. Thus given labeled sets S_{spec} and S_{nspec} the task is to train a model that, for each sentence s, is capable of deciding whether a previously unseen s is speculative or not.

The contributions of this paper are the following:

- The construction of a complex feature selection procedure which successfully reduces the number of keyword candidates without excluding helpful keywords.
- We demonstrate that with a very limited amount of expert supervision in finalising the feature representation, it is possible to build accurate hedge classifiers from (semi-) automatically collected training data.
- The extension of the feature representation used by previous works with bigrams and trigrams and an evaluation of the benefit of using longer keywords in hedge classification.
- We annotated a small test corpora of biomedical scientific papers from a different source to demonstrate that hedge keywords are highly task-specific and thus constructing models that generalise well from one task to another is not feasible without a noticeable loss in accuracy.

2 Methods

2.1 Feature space representation

Hedge classification can essentially be handled by acquiring task specific keywords that trigger speculative assertions more or less independently of each other. As regards the nature of this task, a vector space model (VSM) is a straightforward and suitable representation for statistical learning. As VSM

[3] http://www.computationalmedicine.org/challenge/index.php

is inadequate for capturing the (possibly relevant) relations between subsequent tokens, we decided to extend the representation with bi- and trigrams of words. We chose not to add any weighting of features (by frequency or importance) and for the Maximum Entropy Model classifier we included binary data about whether single features occurred in the given context or not.

2.2 Probabilistic training data acquisition

To build our classifier models, we used the dataset gathered and made available by (Medlock and Briscoe, 2007). They commenced with the seed set S_{spec} gathered automatically (all sentences containing *suggest* or *likely* – two very good speculative keywords), and S_{nspec} that consisted of randomly selected sentences from which the most probable speculative instances were filtered out by a pattern matching and manual supervision procedure. With these seed sets they then performed the following iterative method to enlarge the initial training sets, adding examples to both classes from an unlabelled pool of sentences called U:

1. Generate seed training data: S_{spec} and S_{nspec}
2. Initialise: $T_{spec} \leftarrow S_{spec}$ and $T_{nspec} \leftarrow S_{nspec}$
3. Iterate:
   - Train classifier using T_{spec} and T_{nspec}
   - Order U by $P(spec)$ values assigned by the classifier
   - $T_{spec} \leftarrow$ most probable batch
   - $T_{nspec} \leftarrow$ least probable batch

What makes this iterative method efficient is that, as we said earlier, hedging is expressed via keywords in natural language texts; and often several keywords are present in a single sentence. The seed set S_{spec} contained either *suggest* or *likely*, and due to the fact that other keywords cooccur with these two in many sentences, they appeared in S_{spec} with reasonable frequency. For example, $P(spec|may) = 0.9985$ on the seed sets created by (Medlock and Briscoe, 2007). The iterative extension of the training sets for each class further boosted this effect, and skewed the distribution of speculative indicators as sentences containing them were likely to be added to the extended training set for the speculative class, and unlikely to fall into the non-speculative set.

We should add here that the very same feature has an inevitable, but very important side effect that is detrimental to the classification accuracy of models trained on a dataset which has been obtained this way. This side effect is that other words (often common words or stopwords) that tend to cooccur with hedge cues will also be subject to the same iterative distortion of their distribution in speculative and non-speculative uses. Perhaps the best example of this is the word *it*. Being a stopword in our case, and having no relevance at all to speculative assertions, it has a class conditional probability of $P(spec|it) = 74.67\%$ on the seed sets. This is due to the use of phrases like *it suggests that*, *it is likely*, and so on. After the iterative extension of training sets, the class-conditional probability of *it* dramatically increased, to $P(spec|it) = 94.32\%$. This is a consequence of the frequent co-occurence of *it* with meaningful hedge cues and the probabilistic model used and happens with many other irrelevant terms (not just stopwords). The automatic elimination of these irrelevant candidates is one of our main goals (to limit the number of candidates for manual consideration and thus to reduce the human effort required to select meaningful hedge cues).

This shows that, in addition to the desired effect of introducing further speculative keywords and biasing their distribution towards the speculative class, this iterative process also introduces significant noise into the dataset. This observation led us to the conclusion that in order to build efficient classifiers based on this kind of dataset, we should filter out noise. In the next part we will present our feature selection procedure (evaluated in the Results section) which is capable of underranking irrelevant keywords in the majority of cases.

2.3 Feature (or keyword) selection

To handle the inherent noise in the training dataset that originates from its weakly supervised construction, we applied the following feature selection procedure. The main idea behind it is that it is unlikely that more than two keywords are present in the text, which are useful for deciding whether an instance is speculative. Here we performed the following steps:

1. We ranked the features x by frequency and their class conditional probability $P(spec|x)$. We then selected those features that had $P(spec|x) > 0.94$ (this threshold was chosen arbitrarily) and appeared in the training dataset with reasonable frequency (frequency above 10^{-5}). This set constituted the 2407 candidates which we used in the second analysis phase.

2. For trigrams, bigrams and unigrams – processed separately – we calculated a new class-conditional probability for each feature x, discarding those observations of x in speculative instances where x was not among the two highest ranked candidate. Negative credit was given for all occurrences in non-speculative contexts. We discarded any feature that became unreliable (i.e. any whose frequency dropped below the threshold or the strict class-conditional probability dropped below 0.94). We did this separately for the uni-, bi- and trigrams to avoid filtering out longer phrases because more frequent, shorter candidates took the credit for all their occurrences. In this step we filtered out 85% of all the keyword candidates and kept 362 uni-, bi-, and trigrams altogether.

3. In the next step we re-evaluated all 362 candidates together and filtered out all phrases that had a shorter and thus more frequent substring of themselves among the features, with a similar class-conditional probability on the speculative class (worse by 2% at most). Here we discarded a further 30% of the candidates and kept 253 uni-, bi-, and trigrams altogether.

This efficient way of reranking and selecting potentially relevant features (we managed to discard 89.5% of all the initial candidates automatically) made it easier for us to manually validate the remaining keywords. This allowed us to incorporate supervision into the learning model in the feature representation stage, but keep the weakly supervised modelling (with only 5 minutes of expert supervision required).

2.4 Maximum Entropy Classifier

Maximum Entropy Models (Berger et al., 1996) seek to maximise the conditional probability of classes, given certain observations (features). This is performed by weighting features to maximise the likelihood of data and, for each instance, decisions are made based on features present at that point, thus maxent classification is quite suitable for our purposes. As feature weights are mutually estimated, the maxent classifier is capable of taking feature dependence into account. This is useful in cases like the feature *it* being dependent on others when observed in a speculative context. By downweighting such features, maxent is capable of modelling to a certain extent the special characteristics which arise from the automatic or weakly supervised training data acquisition procedure. We used the OpenNLP maxent package, which is freely available[4].

3 Results

In this section we will present our results for hedge classification as a standalone task. In experiments we made use of the hedge classification dataset of scientific texts provided by (Medlock and Briscoe, 2007) and used a labeled dataset generated automatically based on false positive predictions of an ICD-9-CM coding system.

3.1 Results for hedge classification in biomedical texts

As regards the degree of human intervention needed, our classification and feature selection model falls within the category of weakly supervised machine learning. In the following sections we will evaluate our above-mentioned contributions one by one, describing their effects on feature space size (efficiency in feature and noise filtering) and classification accuracy. In order to compare our results with Medlock and Briscoe's results (Medlock and Briscoe, 2007), we will always give the $BEP(spec)$ that they used – the break-even-point of precision and recall[5]. We will also present $F_{\beta=1}(spec)$ values

[4] `http://maxent.sourceforge.net/`

[5] It is the point on the precision-recall curve of *spec* class where $P = R$. If an exact $P = R$ cannot be realised due to the equal ranking of many instances, we use the point closest to $P = R$ and set $BEP(spec) = (P + R)/2$. BEP is an

which show how good the models are at recognising speculative assertions.

3.1.1 The effects of automatic feature selection

The method we proposed seems especially effective in the sense that we successfully reduced the number of keyword candidates from an initial 2407 words having $P(spec|x) > 0.94$ to 253, which is a reduction of almost 90%. During the process, very few useful keywords were eliminated and this indicated that our feature selection procedure was capable of distinguishing useful keywords from noise (i.e. keywords having a very high speculative class-conditional probability due to the skewed characteristics of the automatically gathered training dataset). The 2407-keyword model achieved a $BEP(spec)$ os 76.05% and $F_{\beta=1}(spec)$ of 73.61%, while the model after feature selection performed better, achieving a $BEP(spec)$ score of 78.68% and $F_{\beta=1}(spec)$ score of 78.09%. Simplifying the model to predict a $spec$ label each time a keyword was present (by discarding those 29 features that were too weak to predict $spec$ alone) slightly increased both the $BEP(spec)$ and $F_{\beta=1}(spec)$ values to 78.95% and 78.25%. This shows that the Maximum Entropy Model in this situation could not learn any meaningful hypothesis from the cooccurence of individually weak keywords.

3.1.2 Improvements by manual feature selection

After a dimension reduction via a strict reranking of features, the resulting number of keyword candidates allowed us to sort the retained phrases manually and discard clearly irrelevant ones. We judged a phrase irrelevant if we could consider no situation in which the phrase could be used to express hedging. Here 63 out of the 253 keywords retained by the automatic selection were found to be **potentially** relevant in hedge classification. All these features were sufficient for predicting the $spec$ class alone, thus we again found that the learnt model reduced to a single keyword-based decision.[6] These 63 keywords yielded a classifier with a $BEP(spec)$ score of 82.02% and $F_{\beta=1}(spec)$ of 80.88%.

3.1.3 Results obtained adding external dictionaries

In our final model we added the keywords used in (Light et al., 2004) and those gathered for our ICD-9-CM hedge detection module. Here we decided not to check whether these keywords made sense in scientific texts or not, but instead left this task to the maximum entropy classifier, and added only those keywords that were found reliable enough to predict $spec$ label alone by the maxent model trained on the training dataset. These experiments confirmed that hedge cues are indeed task specific – several cues that were reliable in radiology reports proved to be of no use for scientific texts. We managed to increase the number of our features from 63 to 71 using these two external dictionaries.

These additional keywords helped us to increase the overall coverage of the model. Our final hedge classifier yielded a $BEP(spec)$ score of 85.29% and $F_{\beta=1}(spec)$ score of 85.08% (89.53% Precision, 81.05% Recall) for the speculative class. This meant an overall classification accuracy of 92.97%.

Using this system as a pre-processing module for a hypothetical gene interaction extraction system, we found that our classifier successfully excluded gene names mentioned in a speculative sentence (it removed 81.66% of all speculative mentions) and this filtering was performed with a respectable precision of 93.71% ($F_{\beta=1}(spec) = 87.27\%$).

Articles	4
Sentences	1087
Spec sentences	190
Nspec sentences	897

Table 1: Characteristics of the BMC hedge dataset.

3.1.4 Evaluation on scientific texts from a different source

Following the annotation standards of Medlock and Briscoe (Medlock and Briscoe, 2007), we manually annotated 4 full articles downloaded from the

interesting metric as it demonstrates how well we can trade-off precision for recall.

[6] We kept the test set blind during the selection of relevant keywords. This meant that some of them eventually proved to be irrelevant, or even lowered the classification accuracy. Examples of such keywords were *will*, *these data* and *hypothesis*. We assumed that these might suggest a speculative assertion.

BMC Bioinformatics website to evaluate our final model on documents from an external source. The chief characteristics of this dataset (which is available at[7]) is shown in Table 1. Surprisingly, the model learnt on FlyBase articles seemed to generalise to these texts only to a limited extent. Our hedge classifier model yielded a $BEP(spec) = 75.88\%$ and $F_{\beta=1}(spec) = 74.93\%$ (mainly due to a drop in precision), which is unexpectedly low compared to the previous results.

Analysis of errors revealed that some keywords which proved to be very reliable hedge cues in FlyBase articles were also used in non-speculative contexts in the BMC articles. Over 50% (24 out of 47) of our false positive predictions were due to the different use of 2 keywords, *possible* and *likely*. These keywords were many times used in a mathematical context (referring to probabilities) and thus expressed no speculative meaning, while such uses were not represented in the FlyBase articles (otherwise bigram or trigram features could have captured these non-speculative uses).

3.1.5 The effect of using 2-3 word-long phrases as hedge cues

Our experiments demonstrated that it is indeed a good idea to include longer phrases in the vector space model representation of sentences. One third of the features used by our advanced model were either bigrams or trigrams. About half of these were the kind of phrases that had no unigram components of themselves in the feature set, so these could be regarded as meaningful standalone features. Examples of such speculative markers in the fruit fly dataset were: *results support, these observations, indicate that, not clear, does not appear, ...* The majority of these phrases were found to be reliable enough for our maximum entropy model to predict a speculative class based on that single feature.

Our model using just unigram features achieved a $BEP(spec)$ score of 78.68% and $F_{\beta=1}(spec)$ score of 80.23%, which means that using bigram and trigram hedge cues here significantly improved the performance (the difference in $BEP(spec)$ and $F_{\beta=1}(spec)$ scores were 5.23% and 4.97%, respectively).

[7] http://www.inf.u-szeged.hu/~szarvas/homepage/hedge.html

3.2 Results for hedge classification in radiology reports

In this section we present results using the above-mentioned methods for the automatic detection of speculative assertions in radiology reports. Here we generated training data by an automated procedure. Since hedge cues cause systems to predict false positive labels, our idea here was to train Maximum Entropy Models for the false positive classifications of our ICD-9-CM coding system using the vector space representation of radiology reports. That is, we classified every sentence that contained a medical term (disease or symptom name) and caused the automated ICD-9 coder[8] to predict a false positive code was treated as a speculative sentence and all the rest were treated as non-speculative sentences.

Here a significant part of the false positive predictions of an ICD-9-CM coding system that did not handle hedging originated from speculative assertions, which led us to expect that we would have the most hedge cues among the top ranked keywords which implied false positive labels.

Taking the above points into account, we used the training set of the publicly available ICD-9-CM dataset to build our model and then evaluated each single token by this model to measure their predictivity for a false positive code. Not surprisingly, some of the best hedge cues appeared among the highest ranked features, while some did not (they did not occur frequently enough in the training data to be captured by statistical methods).

For this task, we set the initial $P(spec|x)$ threshold for filtering to 0.7 since the dataset was generated by a different process and we expected hedge cues to have lower class-conditional probabilities without the effect of the probabilistic data acquisition method that had been applied for scientific texts. Using all 167 terms as keywords that had $P(spec|x) > 0.7$ resulted in a hedge classifier with an $F_{\beta=1}(spec)$ score of 64.04%

After the feature selection process 54 keywords were retained. This 54-keyword maxent classifier got an $F_{\beta=1}(spec)$ score of 79.73%. Plugging this model (without manual filtering) into the ICD-9 coding system as a hedge module, the ICD-9 coder

[8] Here the ICD-9 coding system did not handle the hedging task.

yielded an F measure of 88.64%, which is much better than one without a hedge module (79.7%).

Our experiments revealed that in radiology reports, which mainly concentrate on listing the identified diseases and symptoms (facts) and the physician's impressions (speculative parts), detecting hedge instances can be performed accurately using unigram features. All bi- and trigrams retained by our feature selection process had unigram equivalents that were eliminated due to the noise present in the automatically generated training data.

We manually examined all keywords that had a $P(spec) > 0.5$ given as a standalone instance for our maxent model, and constructed a dictionary of hedge cues from the promising candidates. Here we judged 34 out of 54 candidates to be potentially useful for hedging. Using these 34 keywords we got an $F_{\beta=1}(spec)$ performance of 81.96% due to the improved precision score.

Extending the dictionary with the keywords we gathered from the fruit fly dataset increased the $F_{\beta=1}(spec)$ score to 82.07% with only one out-domain keyword accepted by the maxent classifier.

	Biomedical papers		Medical reports
	$BEP(spec)$	$F_{\beta=1}(spec)$	$F_{\beta=1}(spec)$
Baseline 1	60.00	–	48.99
Baseline 2	76.30	–	–
All features	76.05	73.61	64.04
Feature selection	78.68	78.09	79.73
Manual feat. sel.	82.02	80.88	81.96
Outer dictionary	85.29	85.08	82.07

Table 2: Summary of results.

4 Conclusions

The overall results of our study are summarised in a concise way in Table 2. We list $BEP(spec)$ and $F_{\beta=1}(spec)$ values for the scientific text dataset, and $F_{\beta=1}(spec)$ for the clinical free text dataset. Baseline 1 denotes the substring matching system of Light et al. (Light et al., 2004) and Baseline 2 denotes the system of Medlock and Briscoe (Medlock and Briscoe, 2007). For clinical free texts, Baseline 1 is an out-domain model since the keywords were collected for scientific texts by (Light et al., 2004). The third row corresponds to a model using all keywords $P(spec|x)$ above the threshold and the fourth row a model after automatic noise filtering, while the fifth row shows the performance after the manual filtering of automatically selected keywords. The last row shows the benefit gained by adding reliable keywords from an external hedge keyword dictionary.

Our results presented above confirm our hypothesis that speculative language plays an important role in the biomedical domain, and it should be handled in various NLP applications. We experimentally compared the general features of this task in texts from two different domains, namely medical free texts (radiology reports), and scientific articles on the fruit fly from FlyBase.

The radiology reports had mainly unambiguous single-term hedge cues. On the other hand, it proved to be useful to consider bi- and trigrams as hedge cues in scientific texts. This, and the fact that many hedge cues were found to be ambiguous (they appeared in both speculative and non-speculative assertions) can be attributed to the literary style of the articles. Next, as the learnt maximum entropy models show, the hedge classification task reduces to a lookup for single keywords or phrases and to the evaluation of the text based on the most relevant cue alone. Removing those features that were insufficient to classify an instance as a hedge individually did not produce any difference in the $F_{\beta=1}(spec)$ scores. This latter fact justified a view of ours, namely that during the construction of a statistical hedge detection module for a given application the main issue is to find the task-specific keywords.

Our findings based on the two datasets employed show that automatic or weakly supervised data acquisition, combined with automatic and manual feature selection to eliminate the skewed nature of the data obtained, is a good way of building hedge classifier modules with an acceptable performance.

The analysis of errors indicate that more complex features like dependency structure and clausal phrase information could only help in allocating the scope of hedge cues detected in a sentence, not the detection of any itself. Our finding that token unigram features are capable of solving the task accurately agrees with the the results of previous works on hedge classification ((Light et al., 2004), (Med-

lock and Briscoe, 2007)), and we argue that 2-3 word-long phrases also play an important role as hedge cues and as non-speculative uses of an otherwise speculative keyword as well (i.e. to resolve an ambiguity). In contrast to the findings of Wiebe et al. ((Wiebe et al., 2004)), who addressed the broader task of subjectivity learning and found that the density of other potentially subjective cues in the context benefits classification accuracy, we observed that the co-occurence of speculative cues in a sentence does not help in classifying a term as speculative or not. Realising that our learnt models never predicted speculative labels based on the presence of two or more individually weak cues and discarding such terms that were not reliable enough to predict a speculative label (using that term alone as a single feature) slightly improved performance, we came to the conclusion that even though speculative keywords tend to cooccur, and two keywords are present in many sentences; hedge cues have a speculative meaning (or not) on their own without the other term having much impact on this.

The main issue thus lies in the selection of keywords, for which we proposed a procedure that is capable of reducing the number of candidates to an acceptable level for human evaluation – even in data collected automatically and thus having some undesirable properties.

The worse results on biomedical scientific papers from a different source also corroborates our finding that hedge cues can be highly ambiguous. In our experiments two keywords that are practically never used in a non-speculative context in the FlyBase articles we used for training were responsible for 50% of false positives in BMC texts since they were used in a different meaning. In our case, the keywords *possible* and *likely* are apparently always used as speculative terms in the FlyBase articles used, while the articles from BMC Bioinformatics frequently used such cliche phrases as *all possible combinations* or *less likely / more likely* ... (referring to probabilities shown in the figures). This shows that the portability of hedge classifiers is limited, and cannot really be done without the examination of the specific features of target texts or a more heterogenous corpus is required for training. The construction of hedge classifiers for each separate target application in a weakly supervised way seems feasible though. Collecting bi- and trigrams which cover non-speculative usages of otherwise common hedge cues is a promising solution for addressing the false positives in hedge classifiers and for improving the portability of hedge modules.

4.1 Resolving the scope of hedge keywords

In this paper we focused on the recognition of hedge cues in texts. Another important issue would be to determine the scope of hedge cues in order to locate uncertain sentence parts. This can be solved effectively using a parser adapted for biomedical papers. We manually evaluated the parse trees generated by (Miyao and Tsujii, 2005) and came to the conclusion that for each keyword it is possible to define the scope of the keyword using subtrees linked to the keyword in the predicate-argument syntactic structure or by the immediate subsequent phrase (e.g. prepositional phrase). Naturally, parse errors result in (slightly) mislocated scopes but we had the general impression that state-of-the-art parsers could be used efficiently for this issue. On the other hand, this approach requires a human expert to define the scope for each keyword separately using the predicate-argument relations, or to determine keywords that act similarly and their scope can be located with the same rules. Another possibility is simply to define the scope to be each token up to the end of the sentence (and optionally to the previous punctuation mark). The latter solution has been implemented by us and works accurately for clinical free texts. This simple algorithm is similar to NegEx (Chapman et al., 2001) as we use a list of phrases and their context, but we look for punctuation marks to determine the scopes of keywords instead of applying a fixed window size.

Acknowledgments

This work was supported in part by the NKTH grant of Jedlik Ányos R&D Programme 2007 of the Hungarian government (codename TUDORKA7). The author wishes to thank the anonymous reviewers for valuable comments and Veronika Vincze for valuable comments in linguistic issues and for help with the annotation work.

References

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 5:301–310.

Ken Hyland. 1994. Hedging in academic writing and eap textbooks. *English for Specific Purposes*, 13(3):239–256.

Marc Light, Xin Ying Qiu, and Padmini Srinivasan. 2004. The language of bioscience: Facts, speculations, and statements in between. In Lynette Hirschman and James Pustejovsky, editors, *HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases*, pages 17–24, Boston, Massachusetts, USA, May 6. Association for Computational Linguistics.

Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 992–999, Prague, Czech Republic, June. Association for Computational Linguistics.

Yusuke Miyao and Jun'ichi Tsujii. 2005. Probabilistic disambiguation models for wide-coverage HPSG parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 83–90, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Marie A. Moisio. 2006. *A Guide to Health Insurance Billing*. Thomson Delmar Learning.

John P. Pestian, Chris Brew, Pawel Matykiewicz, DJ Hovermale, Neil Johnson, K. Bretonnel Cohen, and Wlodzislaw Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing*, pages 97–104, Prague, Czech Republic, June. Association for Computational Linguistics.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the Seventh Computational Natural Language Learning Conference*, pages 25–32, Edmonton, Canada, May-June. Association for Computational Linguistics.

James G. Shanahan, Yan Qu, and Janyce Wiebe. 2005. *Computing Attitude and Affect in Text: Theory and Applications (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Janyce Wiebe, Theresa Wilson, Rebecca F. Bruce, Matthew Bell, and Melanie Martin. 2004. Learning subjective language. *Computational Linguistics*, 30(3):277–308.