A Comparison of Classifiers for Detecting Hedges

Sin-Jae Kang[1], In-Su Kang[2], and Seung-Hoon Na[3]

[1] School of Computer and Information Technology, Daegu University
Gyeonsan. Gyeongbuk. 712-714 South Korea
sjkang@daegu.ac.kr

[2] School of Computer Science & Engineering, Kyungsung University
Pusan. 608-736 South Korea
dbaisk@ks.ac.kr

[3] Department of Computer Science, National University of Singapore
nash@comp.nus.edu.sg

Abstract. A hedge is a linguistic device used to avoid using a categorical sentence. Hedges can be used to determine whether a sentence is factual by merely regarding a sentence containing hedges as non-factual. In this paper, we perform a comparative experiment of various classification methods for hedge detection. Among four different classification methods, we observe that SVM shows the best performance and that the SVM-based method finally outperforms the best system in the CoNLL2010-ST task.

Keywords: Hedge Detection, Machine Learning, Natural Language Processing, Information Extraction.

1 Introduction

A hedge is a linguistic device used to avoid using a categorical sentence [1, 2]. Examples of hedges are *may*, *probably*, *it appears that*, etc. Hedges are used in a sentence when the writer is uncertain or has doubt about the contents of the sentence. Due to this uncertainty, sentences with hedges are considered to be speculative or non-factual.

Motivated by this characteristic of hedges, researchers have exploited hedges as useful features to extract factual sentences from documents. There are many applications which need to determine whether a sentence is factual or not. For example, in designing a specific type of question answering system which finds a factual answer, non-factual parts from documents need to be filtered out to effectively find an answer.

In this paper, we address the problem of hedge detection, i.e., to determine whether or not a sentence contains hedges. As hedge detection generally belongs to a classification problem, all supervised machine learning (ML) approaches can be applied to this problem. However, there has been no previous work on comparing existing methods, thus it is not clear what the best ML approach is for hedge detection. To draw a useful conclusion to this issue, we attempt to compare the state-of-the-art ML approaches for hedge detection. For the comparison, we selected CRF, SVM, *k*-NN, and DT (decision tree learner), as they have been widely used for many classification problems. In our comparison results, SVM shows the best performance among four different classification methods and the SVM-based method finally outperforms the best system in CoNLL2010-ST task.

The paper is organized as follows. In Section 2, we summarize related work. In Section 3, we describe features selected to detect hedges. Experimental results are presented and discussed in Section 4. Finally, Section 5 puts forward some conclusions.

2 Relevant Work

Light et al. [3] used a simple technique, which determines hedge sentences relying on the presence of hedge cue words such as *suggest*, *potential*, *likely*, *may*, etc. There have been a few successful works on gathering a variety of hedge cue terms based on the weakly-supervised learning with SVM [4] or MaxEnt [5]. Such methods start from initial seeds such as *suggest* and *likely*, and extend the set of cue terms by exploiting the assumption that hedge cue words may co-occur with *′may suggest′*. Morante and Daelemans [6] reported 84.7% in F1 by using *k*-NN learning based on the features such as lemmas, words, POS, and IOB tags of its preceding and subsequent three tokens for each token in training sentences.

Recently, CoNLL 2010 Shared Task (CoNLL2010-ST)[1] evaluated the performance of detecting hedge cues and their linguistic scope in biological literatures and Wikipedia documents [7]. Its top-ranked systems adopted a sequence labeling (SL) approach [8-10] based on CRF or SVM_hmm.

Tang et al. proposed a cascaded system of CRF and SVM_hmm. However, the cascaded method was not better than that of their single CRF classifier, which achieved the best performance of 86.79% in the biological task of CoNLL2010-ST [8]. Tang et al. used words, lemmas, POS, chunk tags, and some composite features. Li et al. devised a greedy-forward feature selection scheme to obtain high precision, and showed F-measure of 85.89% using CRF [9]. Zhou et al. exploited the synonym features from WordNet, and showed F-measure of 86.32% based on CRF [10]. In summary, most top ranked systems employed CRF.

3 Features for Detecting Hedges

In order to fairly compare machine learning techniques, the same set of features should be used during the process of machine learning. The commonly-used features in the top ranked systems of CoNLL2010-ST [7] are given as follows.

(1) Word Feature: *word(i)* (i= -2, -1, 0, +1, +2)
(2) Lemma Feature: *lemma(i)* (i= -2, -1, 0, +1, +2)
(3) Part-Of-Speech (POS) Feature: *POS(i)* (i= -2, -1, 0, +1, +2)
(4) Chunk Tag Feature: *chunk(i)* (i= -2, -1, 0, +1, +2)

[1] http://www.inf.u-szeged.hu/rgai/conll2010st/

In the above, *word(0)*, *word(-n)*, and *word(+n)* means respectively the current word, the *n*-th word to the left of the current word, and the *n*-th word to the right of the current word. *lemma(i)* is the stemming result of word(i). *POS(i)* is the part-of-speech of word(i). *chunk(i)* is the chunk tag of word(i), which are represented in the IOB2 format (B for BEGIN, I for INSIDE, and O for OUTSIDE) [11].

4 Experimental Results

As a test set to evaluate several ML approaches to hedge detection, this study uses the biological part of the CoNLL2010-ST dataset which contains biological abstracts and full articles from the BioScope (biomedical domain) corpus. Table 1 shows some statistics of the test set.

Similar to earlier works, this study adopted the following general steps for hedge sentence classification. First, for a given sentence, each token of the sentence is classified into hedge-class or non-hedge-class. The classes are represented in the

Table 1. CoNLL2010-ST biological dataset

	# of training sentences	# of testing sentences
Hedge	2,620	790
Non-hedge	11,921	4,213
Total	14,541	5,003

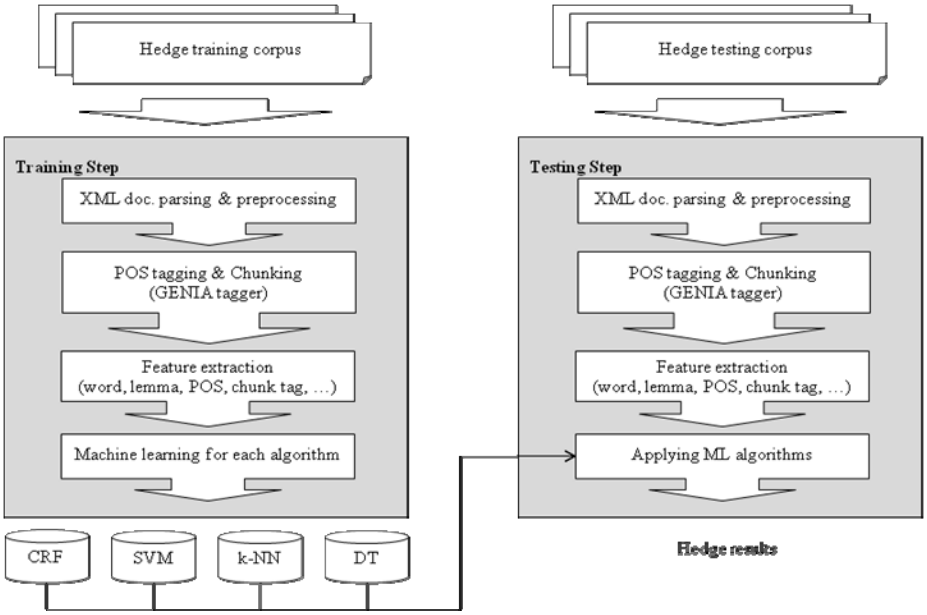
Hedge training corpus
Hedge testing corpus
Training Step
XML doc. parsing & preprocessing
POS tagging & Chunking (GENIA tagger)
Feature extraction (word, lemma, POS, chunk tag, ...)
Machine learning for each algorithm
Testing Step
XML doc. parsing & preprocessing
POS tagging & Chunking (GENIA tagger)
Feature extraction (word, lemma, POS, chunk tag, ...)
Applying ML algorithms
Hedge results
CRF
SVM
k-NN
DT

Fig. 1. The overall process for detecting hedges

IOB2 format (B for BEGIN, I for INSIDE, and O for OUTSIDE of hedge cue words) in order to annotate the cue phrases and the left and right boundaries of their scopes. Then, if there are one or more hedge-class tokens in a sentence, the sentence is classified into a hedge sentence.

The overall training and testing processes for detecting hedges are presented in Fig. 1. The training data has the information about locations of hedge cue words as well as whether each sentence is a hedge sentence or not. Basically, the training and testing step generate a list of raw sentences. POS tagging and chunking are performed against the raw sentences using the GENIA tagger which is known to work well on various types of biomedical documents [12]. From the output of GENIA tagger, the features defined in Section 3 are extracted, and then each classifier of CRF[2], SVM[3], *k*-NN[4], and DT[5] is learned from the training data and evaluated using the test data.

This study evaluated two baseline methods. The first baseline system *Cue-dic* tries to detect hedge sentences solely relying on the cue-term list gathered from the train set. The second baseline system *Prob* finds hedge sentences based on the average hedge-probability of words in a sentence. Suppose that a sentence *S* consists of a sequence of words w_1, w_2, …, w_n. Then, the system *Prob* computes the hedge-probability *Score*(*S*) of *S* which is defined using *Pr*(w_i) the probability that word w_i appears in the hedge sentence. Then, the input sentence *S* is classified into the hedge class if *Score*(*S*) is over a threshold.

$$\Pr(w_i) = \frac{\textit{frequency of } w_i \textit{ in hedge sentences of training corpus}}{\textit{frequency of } w_i \textit{ in training corpus}} \quad (1)$$

$$Score(S) = \left(\sum_{i=1}^{n} \Pr(w_i) \right) \Big/ n \quad (2)$$

Table 2 shows the evaluation results of six methods including four ML approaches which employ the same set of features defined in Section 3. *Cue-dic* used 1,344 hedge cue lists, which consist of 168 one-word cues, 420 two-word cues, and 756 more than three-word cues. In *Prob*, threshold 0.3 showed the best result among values between 0 and 1. For SVM, the 2nd degree of polynomial kernel and 1 slack variable were used. CRF used L2 for the regularization algorithm, 1.5 for hyper-parameter, and 1 for cut-off threshold for the features. In *k*-NN, when k is 2, *k*-NN showed the best result among k values between 1 and 4. DT used IB1 algorithm, weighted overlap for feature metrics, and GainRatio weight.

The result of *Cue-dic* indicates that a set of known hedge terms recalls most hedge sentences, but hedge cue terms appear nearly and equally in either hedge sentences or non-hedge ones. Interestingly, *Prob* method well discriminated hedge sentences, performing slightly better than or comparably to the two ML methods k-NN and DT.

[2] http://crfpp.sourceforge.net/

[3] http://chasen.org/~taku/software/yamcha/#rpm

[4] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[5] http://www.rulequest.com/Personal/

Table 2. The comparion results of hedge detection using the features defined in Section 3

Method		Precision	Recall	F1
Baseline	Cue-dic	48.04	94.56	63.71
	Prob	81.85	85.06	83.43
ML	CRF	87.14	84.05	85.57
	SVM	89.57	82.66	85.98
	k-NN	81.10	83.67	82.37
	DT	82.21	84.81	83.49

In later studies on detecting hedges, *Prob* could thus be regarded as a strong baseline approach.

Expectedly, SVM and CRF performed better than k-NN and DT, as reported in the previous studies. SVM and CRF showed greater precision than recall, while k-NN and DT made the opposite result. Unlike the results of the top-performing systems in CoNLL2010-ST, SVM achieved better effectiveness than CRF in this experiment, although the difference was marginal. Actually, two of the top-3 teams in CoNLL2010-ST attempted both HM-SVM and CRF to find the better. HM-SVM (Hidden Markov SVM) is a combination of HMM and SVM proposed for label sequence learning [8-9]. In our evaluation, however, SVM classifies each token in a sentence independently of other tokens. This is related to the difference between sequence labeling and token classification approaches to hedge detection. In other words, CRF and SVM in Table 2 can be viewed as respectively the representative methods of sequence labeling and token classification approaches. Sequence label learning like CRF may be more advantageous for hedge detection if the result of hedge detection is used by hedge scoping which should determine the boundaries of hedges in a sentence. Otherwise, however, token classification may be more appropriate in the sense that its learning process could focus on finding the most influential hedge token for improving sentence-level classification effectiveness instead of locating the entire hedge tokens of a single hedge as in sequence labeling.

We did additional experiments with SVM which performed best shown in Table 2. In addition to the common essential features defined in Section 3, we added *prev_res(j)* feature, which means the classification result of previous tokens within a window size. As shown in Table 3, SVM equipped with such additional feature led to improvement, yielding 86.82% in F1 as well as slightly outperforming the CoNLL2010-ST best systems. This indicates that sequential features like classification labels of preceding tokens may be helpful in detecting hedges using non-sequential labelers like SVM. As far as we know, previous tag feature like ours has not been successfully explored in hedge detection.

As to features for hedge sentence detection, the previous best systems employed not only well-known sentential contextual features (words, lemmas, POS, and chunks of neighboring words), but also additional complex features such as synonym features, prefix/suffix, etc. For example, Tang [8] used several composite features such as a concatenation of the lemma and chunk of the same token. Unlike earlier state-of-the-art systems, the evaluation result of this study was obtained only from well-known local contextual features. This means that our result can be easily replicated by other researchers for further improvements.

Table 3. The results of hedge detection for different feature sets, using SVM for classifier

Feature set	Precision	Recall	F1
word(i), lemma(i), pos(i), chunk(i), prev_res(j) (-1 ≤ i ≤ 1, -2 ≤ j ≤ -1)	86.79	86.46	86.62
word(i), lemma(i), pos(i), chunk(i), prev_res(j) (-2 ≤ i ≤ 2, -2 ≤ j ≤ -1)	90.07	83.80	86.82
word(i), lemma(i), pos(i), chunk(i), prev_res(j) (-3 ≤ i ≤ 3, -2 ≤ j ≤ -1)	91.60	81.39	86.19
word(i), lemma(i), pos(i), chunk(i) (-2 ≤ i ≤ 2)	89.57	82.66	85.98
word(i), lemma(i), pos(i), chunk(i), prev_res(j) (-2 ≤ i ≤ 2, j = -1)	90.00	83.16	86.45
word(i), lemma(i), pos(i), chunk(i), prev_res(j) (-2 ≤ i ≤ 2, -3 ≤ j ≤ -1)	90.18	83.67	86.80

5 Conclusion

This study compared various machine learning techniques for detecting hedges and defined the feature set to achieve the best performance. Under the same condition, SVM and CRF showed better performances than k-NN and DT. In addition to common essential features such as words, lemmas, POS, and chunks, an additional sequential feature such as classification labels of previous words enabled a further improvement in SVM classifier. Fortunately, this study has found the best ML method and best set of features for hedge detection, which showed a slightly better effectiveness than the previous state-of-the-art hedge detectors.

In the future, we will apply our SVM-based hedge detector to the query answering task and examine the effect of using a hedge detector on the performance of answer extraction.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0007025).

References

1. Lakoff, G.: Hedges: a study in meaning criteria and the logic of fuzzy concepts. Chicago Linguistics Society Papers 8, 183–228 (1972)
2. Hyland, K.: Persuasion, interaction and the construction of knowledge: representing self and others in research writing. International Journal of English Studies 8(2), 8–18 (2008)
3. Light, M., Qiu, X.Y., Srinivasan, P.: The language of bioscience: facts, speculations, and statements in between. In: Proceedings of BioLINK 2004: Linking Biological Literature, Ontologies and Databases, pp. 17–24 (2004)
4. Medlock, B., Briscoe, T.: Weakly supervised learning for hedge classification in scientific literature. In: Proceedings of 45th Meeting of the Association for Computational Linguistics, pp. 992–999 (2007)
5. Szarvas, G.: Hedge classification in biomedical texts with a weakly supervised selection of keywords. In: Proceedings of 46th Meeting of the Association for Computational Linguistics, pp. 281–289 (2008)
6. Morante, R., Daelemans, W.: Learning the scope of hedge cues in biomedical texts. In: Proceedings of the BioNLP 2009 Workshop, pp. 28–36 (2009)
7. Farkas, R., Vincze, V., Mora, G., Csirik, J., Szarvas, G.: The CoNLL 2010 Shared Task: Learning to Detect Hedges and their Scope in Natural Language Text. In: Proceedings of the Shared Task, 14th Conference on Computational Natural Language Learning, Sweden, pp. 1–12 (2010)
8. Tang, B., Wang, X., Yuan, B., Fan, S.: A Cascade Method for Detecting Hedges and their Scope in Natural Language Text. In: Proceedings of the Shared Task, 14th Conference on Computational Natural Language Learning, Sweden, pp. 13–17 (2010)
9. Li, X., Shen, J., Gao, X., Wang, X.: Exploiting Rich Features for Detecting Hedges and Their Scope. In: Proceedings of the Shared Task, 14th Conference on Computational Natural Language Learning, Sweden, pp. 78–83 (2010)
10. Zhou, H., Li, X., Huang, D., Li, Z., Yang, Y.: Exploiting Multi-Features to Detect Hedges and Their Scope in Biomedical Texts. In: Proceedings of the Shared Task, 14th Conference on Computational Natural Language Learning, Sweden, pp. 106–113 (2010)
11. Sang, T.K., Veenstra, J.: Representing Text Chunks. In: Proc. of EACL 1999, pp. 173–179 (1999)
12. Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., Tsujii, J.: Developing a Robust Part-of-Speech Tagger for Biomedical Text. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 382–392. Springer, Heidelberg (2005)