A Baseline Approach for Detecting Sentences Containing Uncertainty

Erik Tjong Kim Sang
University of Groningen
erikt@xs4all.nl

Abstract

We apply a baseline approach to the CoNLL-2010 shared task data sets on hedge detection. Weights have been assigned to cue words marked in the training data based on their occurrences in certain and uncertain sentences. New sentences received scores that correspond with those of their best scoring cue word, if present. The best acceptance scores for uncertain sentences were determined using 10-fold cross validation on the training data. This approach performed reasonably on the shared task's biological (F=82.0) and Wikipedia (F=62.8) data sets.

1 Introduction

CoNLL-2010 offered two shared tasks which involve finding text parts which express uncertainty or unreliability (Farkas et al., 2010). We focus on Task 1, identifying sentences which contain statements which can be considered uncertain or unreliable. We train a basic statistical model on the training data supplied for the task, apply the trained model to the test data and discuss the results. The next section describes the format of the data and introduces the model that was used. Section three discusses the experiments with the model and their results. Section four concludes the paper.

2 Data and model

The CoNLL-2010 shared task training data sets contain sentences which are classified as either *certain* or *uncertain*. Sentences of the uncertain class contain one or more words which have been marked as indicator of uncertainty, the so-called hedge cues. Here is an example of such a sentence with the hedge cues written in **bold** font:

> These results **indicate that** in monocytic cell lineage, HIV-1 **could** mimic some differentiation/activation stimuli allowing nuclear NF-KB expression.

CoNLL-2010 offers two shared tasks: classifying sentences in running text as either *certain* or *uncertain* (Task 1) and finding hedge cues in sentences classified as *uncertain* together with their scopes (Task 2). We have only participated in Task 1.

We built a basic model for the training data, taking advantage of the fact that the hedge cues were marked explicitly. We estimated the probability of each training data word appearing in a hedge cue with unigram statistics:

$$P(w\ in\ cue) = \frac{f(w\ in\ cue)}{f(w)}$$

where $P(w\ in\ cue)$ is the probability that word w appears in a hedge cue, $f(w)$ is frequency of the word w in the data and $f(w\ in\ c)$ is the frequency of the word inside hedge cues. We performed only little text preprocessing, converting all words to lower case and separating six common punctuation signs from the words.

In the classification stage, we assigned to each word the estimated hedge cue probability according to the training data. Next, we assigned a score to each sentence that was equal to one minus the highest individual score of its words:

$$P(s\ is\ certain) = 1 - \operatorname*{argmax}_{w\ in\ s} P(w\ in\ cue)$$

$P(s\ is\ certain)$ is the estimated probability that the sentence s is certain, and it is equal to one minus the highest probability of any of its words being part of a hedge cue. So a sentence containing only words that never appeared as a hedge cue would receive score 1.0. Meanwhile a sentence

Proceedings of the Fourteenth Conference on Computational Natural Language Learning: Shared Task, pages 148–150,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

with a single word that had appeared in a hedge cue in the training data would receive one minus the probability associated with that word. This model ignores any relations between the words of the sentence. We experimented with combining the scores of the different words but found the minimum word score to perform best.

3 Experiments

Apart from the word probabilities, we needed to obtain a good threshold score for deciding whether to classify a sentence as *certain* or *uncertain*. For this purpose, we performed a 10-fold cross-validation experiment on each of the two training data files (biological and Wikipedia) and measured the effect of different threshold values. The results can be found in Figure 1.

The model performed well on the biological training data, with F scores above 80 for a large range of threshold values (0.15–0.85). It performed less well on the Wikipedia training data, with a maximum F score of less than 60 and 50+ scores being limited to the threshold range 0.45–0.85. The maximum F scores were reached for threshold values 0.55 and 0.65 for biological data (F=88.8) and Wikipedia data (F=59.4), respectively. We selected the threshold value 0.55 for our further work because the associated precision and recall values were closer to each other than for value 0.65.

We build domain-specific models with the biological data (14,541 sentences) and the Wikipedia data (11,111 sentences) and applied the models to the related training data. We obtained an F score of 80.2 on the biological data (13th of 20 participants) and a score of 54.4 on the Wikipedia data (9th of 15 participants). The balance between precision and recall scores that we strived for when processing the training data, was not visible in the test results. On the biological test data the system's recall score was 13 points higher than the precision score while on the Wikipedia test data precision outperformed recall by 31 points (see Table 1).

Next, we tested the effect of increasing the data sets with data from another domain. We repeated the cross-validation experiments with the training data, this time adding the available data of the other domain to each of the sets of nine folds used as training data. Unfortunately, this did not result in a performance improvement. The best performance for the biological data dropped to F = 84.2 (threshold 0.60) while the top score for the Wikipedia data dropped to F = 56.5 (0.70).

train-test	thre.	Precis.	Recall	$F_{\beta=1}$
bio-bio	.55	74.3%	87.1%	80.2±1.0
wik-wik	.55	74.0%	43.0%	54.4±0.9
all-bio	.55	69.3%	74.6%	71.8±1.2
all-wik	.55	69.0%	44.6%	54.2±1.0

Table 1: Performances of the models for different combinations of training and test data sets with the associated acceptance threshold values. Training and testing with data from the same domain produces the best scores. Higher recall scores were obtained for biological data than for Wikipedia data. Standard deviations for F scores were estimated with bootstrap resampling (Yeh, 2000).

We kept the threshold value of 0.55, built a model from all available training data and tested its performance on the two test sets. In both cases the performances were lower than the ones obtained with domain dependent training data: F = 71.8 for biological data and F = 54.2 for Wikipedia data (see Table 1).

As post-deadline work, we added statistics for word bigrams to the model, following up work by Medlock (2008), who showed that considering word bigrams had a positive effect on hedge detection. We changed the probability estimation score of words appearing in a hedge cue to

$$P(w_{i-1}w_i\ in\ cue) = \frac{f(w_{i-1}w_i\ in\ cue)}{f(w_{i-1}w_i)}$$

where $w_{i-1}w_i$ is a bigram of successive words in a sentence. Bigrams were considered to be part of a hedge cue when either or both words were inside the hedge cue. Unigram probabilities were used as backoff for known words that appeared outside known bigrams while unknown words received the most common score for known words (0). Sentences received a score which is equal to one minus the highest score of their word bigrams:

$$P(s\,is\,certain) = 1 - \underset{w_{i-1}w_i\ in\ s}{\text{argmax}}\ P(w_{i-1}w_i\,in\,cue)$$

We repeated the threshold estimation experiments and found that new bigram scores enabled the models to perform slightly better on the training

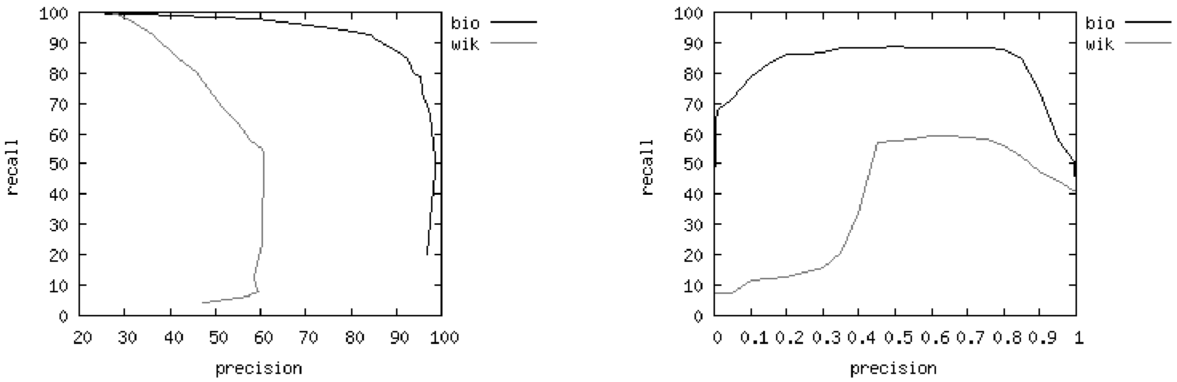

Figure 1: Precision-recall plot (left) and F plot (right) for different values of the certainty acceptance thresholds measured by 10-fold cross-validation experiments on the two shared task training data sets (biological and Wikipedia). The best attained F scores were 88.8 for biological data (threshold 0.55) and 59.4 for Wikipedia data (0.65).

data. The maximum F score for biological training data improved from 88.8 to 90.1 (threshold value 0.35) while the best F score for the Wikipedia training data moved up slightly to 59.8 (0.65).

We applied the bigram models with the two optimal threshold values for the training data to the test data sets. For the biological data, we obtained an F score of 82.0, a borderline significant improvement over the unigram model score. The performance on the Wikipedia data improved significantly, by eight points, to F = 62.8 (see Table 2). This is also an improvement of the official best score for this data set (60.2). We believe that the improvement originates from using the bigram model as well as applying a threshold value that is better suitable for the Wikipedia data set (note that in our unigram experiments we used the same threshold value for all data sets).

4 Concluding remarks

We applied a baseline model to the sentence classification part of the CoNLL-2010 shared task on hedge detection. The model performed reasonably on biological data (F=82.0) but less well on Wikipedia data (F=62.8). The model performed best when trained and tested on data of the same domain. Including additional training data from another domain had a negative effect. Adding bigram statistics to the model, improved its performance on Wikipedia data, especially for recall.

Although the model presented in this paper performs reasonably on the hedge detection tasks, it is probably too simple to outperform more complex models. However, we hope to have shown its usefulness as baseline and as possible feature for more advanced models. We were surprised about the large difference in performance of the model on the two data sets. However, similar performance differences were reported by other participants in the shared task, so they seem data-related rather than being an effect of the chosen model. Finding the origin of the performance differences would be an interesting goal for future work.

train-test	thre.	Precis.	Recall	$F_{\beta=1}$
bio-bio	.35	79.8%	84.4%	82.0±1.1
wik-wik	.65	62.2%	63.5%	62.8±0.8
all-bio	.50	73.2%	77.7%	75.4±1.2
all-wik	.60	63.5%	57.9%	60.6±0.9

Table 2: Performances of bigram models for different combinations of training and test data sets. The bigram models performed better than the unigram models (compare with Table 1).

References

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 *Proc. of the Computational Natural Language Learning (CoNLL-2010): Shared Task*, pages 1–12.

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

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 947–953.