Detecting Semantic Uncertainty by Learning Hedge Cues in Sentences Using an HMM

Xiujun Li
Dept. of Computer Sciences
University of
Wisconsin-Madison
Madison, WI 53706, USA
lixiujun@cs.wisc.edu

Wei Gao
Qatar Computing Research
Institute
Qatar Foundation
Doha, Qatar
wgao@qf.org.qa

Jude W. Shavlik
Dept. of Computer Sciences
University of
Wisconsin-Madison
Madison, WI 53706, USA
shavlik@cs.wisc.edu

ABSTRACT

Detecting speculative assertions is essential to distinguish semantically uncertain information from the factual ones in text. This is critical to the trustworthiness of many intelligent systems that are based on information retrieval and natural language processing techniques, such as question answering or information extraction. We empirically explore three fundamental issues of uncertainty detection: (1) the predictive ability of different learning methods on this task; (2) whether using unlabeled data can lead to a more accurate model; and (3) whether closed-domain training or cross-domain training is better. For these purposes, we adopt two statistical learning approaches to this problem: the commonly used bag-of-words model based on Naive Bayes, and the sequence labeling approach using a Hidden Markov Model (HMM). We empirically compare between our two approaches as well as externally compare with prior results on the CoNLL-2010 Shared Task 1.

Overall, our results are promising: (1) on Wikipedia and biomedical datasets, the HMM model improves over Naive Bayes up to 17.4% and 29.0%, respectively, in terms of absolute F score; (2) compared to CoNLL-2010 systems, our best HMM model achieves 62.9% F score with MLE parameter estimation and 64.0% with EM parameter estimation on Wikipedia dataset, both outperforming the best result (60.2%) of the CoNLL-2010 systems, but our results on the biomedical dataset are less impressive; (3) when the expression ability of a model (e.g., Naive Bayes) is not strong enough, cross-domain training is helpful, and when a model is powerful (e.g., HMM), cross-domain training may produce biased parameters; and (4) under Maximum Likelihood Estimation, combining the unlabeled examples with the labeled helps.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Classifier Design and Evaluation

Keywords

Uncertainty detection, Hedge cues, Naive Bayes, HMM, Cross-domain training

SMIR'14, July 11, 2014, Gold Coast, Queensland, Australia.

1. INTRODUCTION

Speculative language refers to expressions of uncertainty over statements, which indicates that speakers do not back up their opinions with facts. In information retrieval and natural language processing, many applications seek to extract this kind of information and try to distinguish them from the factual information since they convey a different attitude that speakers hold. For example, in question answering (QA), it is of paramount importance to ensure that the candidate answers gathered from various sources are of high certainty or bear sufficient supporting evidence, and those less certain should be automatically pushed downward in the answer list, or to retain users' trust on the QA system, it would be desirable for the system providing the level of uncertainty associated with the output answers [12]. In recent years, with the increasing popularity of social media, the quality of information in terms of factuality becomes a premier concern owing to the casual and word-of-mouth peculiarity of information sources. Uncertainty detection, i.e., distinguishing uncertain statements from factual ones, is becoming increasingly crucial for users to synthesize information to derive reliable interpretations [26].

The problem of uncertainty detection was intensively studied in the CoNLL-2010 shared tasks [4], with two phases: Task 1 is to detect the propositions containing uncertainty at the sentence level, marking the hedge cues if possible; Task 2 is to identify its linguistic scope in sentence. In this paper, we are focused on investigating the solutions for the first task (which is more fundamental). Unlike the individual systems submitted for the shared tasks, we intend to provide a holistic analytics about three different issues that are commonly faced in the development of this kind of systems.

The first issue we explore is the predictive ability of different machine learning models on this task. Typically the basic approach is using a bag-of-words model without considering the correlation among sentence words. For comparison, we formulate the task as a sequence labeling problem to capture the dependency between neighboring words, and employ the classical Hidden Markov Model (HMM) with a specific tag set to label the sentence at the word level. Through the comparison of the two different models, we try to answer whether one can produce more accurate predictions by taking into account word dependencies in representation, and how advantageous the expressiveness of the model is for uncertainty identification.

The second issue is that the commonly employed supervised models suffer from data sparsity problem [9], which is seen strikingly in our dataset. To address this problem, for the Naive Bayes model, we examine different smoothing factors to optimize the parameters; for the HMM we experiment with two parameter estimation methods – Maximum Likelihood Estimation (MLE) and

Expectation Maximization (EM). Under MLE, we provide iterative parameter estimation by incorporating unlabeled data into the training process; under EM [8], we run it for parameter estimation iteratively on the training examples only until convergence.

The CoNLL-2010 Shared Task 1 involves data from two interesting domains: Wikipedia and biomedical scientific literature (both abstracts and full articles). The third issue is that we investigate different kinds of annotated resources in closed-domain training and cross-domain training to see whether we can derive a more accurate model by cross-domain training. Here the cross-domain training is to use the union of Wikipedia and biomedical datasets for training and test on one domain while closed-domain training means that training is done on the same domain as testing.

The remainder of the paper is organized as follows. Section 2 summarizes related work. Section 3 discusses the learning methods we investigated and different parameter estimation methods. Section 4 presents our empirical results, while Section 5 gives our discussion and lessons learned from the experiment. Section 6 concludes the paper.

2. RELATED WORK

In this section, we review some popular uncertainty corpora and methods for uncertainty detection.

Several text corpora from various domains have been annotated over the past few years at different levels (e.g., expression, event, relation, or sentence) with information related to uncertainty detection task. Sauri and Pustejovsky [17] presented a corpus annotated with information about the factuality of events, namely *Factbank*, which is constructed based on *TimeBank*[1] containing 3,123 annotated sentences from 208 news documents with 8 different levels of uncertainty defined. Vincze et al. [24] constructed the BioScope corpus, which consists of medical and biological texts annotated for negation, uncertainty, and their linguistic scope. This corpus contains 20,924 sentences. Ganter et al. [6] generated Wikipedia Weasels Corpus, where *Weasel tags* in Wikipedia articles is adopted readily as labels for uncertainty annotation. It contains 168,923 unique sentences with 437 weasel tags in total. Although several uncertainty corpora exist, there is not a uniform set of standard for uncertainty annotation. Szarvas et al. [19] normalized the annotation of the three corpora aforementioned and provided fine-grained categories of uncertainty (e.g., epistemic, doxastic, investigation, and condition).

Previous work on uncertainty detection focused on classifying sentences into uncertain or definite categories. Existing approaches are mainly based on supervised methods [11, 14, 13, 18] using the annotated corpus with different types of linguistic features including Part-Of-Speech (POS) tags, word stems, n-grams, and so on. Light et al. [11] explore the ability of a Support Vector Machine (SVM) classifier to perform this task on a corps of biomedical abstracts using a stemming representation. Medlock and Briscoe [14] model hedge classification as a weakly supervised machine learning task performed on articles from the functional genomic literature. Medlock [13] presents an extension of this work by experimenting with more features (e.g., POS, stems, and bigrams). Following Medlock and Briscoe [14], Szarvas [18] develops a Maximum Entropy classifier that incorporates bigrams and trigrams in the feature representation and performs a re-ranking based features selection procedure that allows a reduction of the number of keyword candidates from 2,407 to 253.

Later on, classification of uncertain sentences was consolidated as a shared task in CoNLL-2010 on learning to detect hedge cues and their scope in natural language text [4]. The compulsory part is to detect the sentence containing uncertainty at sentence level, while it is optional to mark cues for the uncertainty within the sentence. The approaches on this task fall into two major categories: one is to conduct the binary classification at the sentence level with no cue identification; the other approach is token-level classification to identify whether each token is a part of cue phrase, and then predict the sentence as certainty or uncertainty by counting whether there is any cue phrase in the sentence. The CoNLL report [4] summarized the submissions. A number of models were applied by the submissions, including SVM [7, 20], Conditional Random Fields (CRF) [21, 27], Maximum Entropy [2], k-Nearest Neighbors [15], Naive Bayes [22], Averaged Perceptron [10], and Logistic Regression [25]. For high performance, most of the submissions used a cue dictionary in their system for classification and cue annotation in sentence. Among them, most were conducted on close-domain training, however a few of them employed cross-domain training in their systems, such as Zhao et al. [27] on the biomedical test dataset, and Ji et al. [10] on the Wikipedia test dataset. The best system for Wikipedia data [7] employed SVM, and the best system for biological data [21] adopted CRF.

As a follow-up of the CoNLL shared Task, Velldal [23] proposed to handle the hedge detection task as a simple disambiguation problem, restricted to the words that have previously been observed as hedge cues.

Our approach has three major difference from previous work: (1) usually the cue-based annotation applied a dictionary approach, the key step is to locate the lexical cues, while in our work, we do not use the dictionary and our approach is based on pure statistical models; (2) we used HMM to model the correlation among sentence words; and (3) as for the training resources, we examine both closed-domain and cross-domain training.

3. OUR APPROACH

We approach the task in two ways. One is a bag-of-words approach based on unigrams, in which we perform Naive Bayes classification on the sentence level with no cue identification. The other is sequence labeling, for which we use an HMM with cue annotation in sentences.

3.1 Dataset

We used the standard BioScope corpus as our training and test datasets. On the BioScope corpus, there are two domains. One is a biomedical dataset and the other is a Wikipedia dataset. Both domains are provided with labeled training examples and have hedge cues annotation inside the sentences, plus an unlabeled test set. CoNLL-2010 also provided a labeled test set containing labeled cue information for evaluation. Table 1 gives some samples of annotations in each domain of the corpus. Wikipedia dataset may share some common cue words with the biomedical dataset, while the biomedical context, due to its scientific nature, may have more rigorous structure for uncertain information in sentence. The datasets are freely available at `http://www.inf.u-szeged.hu/rgai/conll2010st/`.

3.2 Design

The aim of Task 1 is to develop automatic procedures for identifying sentences that contain unreliable or uncertain information. This can be formalized as a problem of binary classification (i.e., certain versus uncertain) from the sentence level. As a bag-of-words solution, we built a Naive Bayes classifier based on the unigrams from the training examples, to predict each one of test sen-

[1] `http://www.timeml.org/site/timebank/timebank.html`

Table 1: Example training instances in the corpus from two different domains, where Wikipedia dataset may share some common cue words with the biomedical dataset, while the biomedical context may have more rigorous structure for uncertain information in sentence.

Wikipedia dataset

- <sentence certainty="uncertain" id="S8.2"><*ccue*>*Relatively*</*ccue*> little is known about the early settlement of much of South America east of the Andes.</sentence>
- <sentence certainty="uncertain" id="S25.2">Predatory pricing practices <*ccue*>*may*</*ccue*> result in antitrust claims of monopolization or attempts to monopolize.</sentence>
- <sentence certainty="uncertain" id="S39.2"><*ccue*>*It is not yet clear*</*ccue*></sentence>

Biomedical dataset

Abstracts:

- <sentence certainty="uncertain" id="S76.2">Genes actively involved in the G0/G1 switch (G0S genes) <*ccue*>*may*</*ccue*> be differentially expressed during the lectin-induced switch of lymphocytes from the G0 to the G1 phases of the cell cycle.</sentence>
- <sentence certainty="uncertain" id="S109.1">The AP-1 site at -150 bp, but not the NF-kappa B site, is <*ccue*>*likely*</*ccue*> to represent the major target of protein kinase C in the interleukin 2 promoter.</sentence>
- <sentence certainty="uncertain" id="S113.1">Modulation of normal erythroid differentiation by the endogenous thyroid hormone and retinoic acid receptors: a <*ccue*>*possible*</*ccue*> target for verbA oncogene action.</sentence>

Full articles:

- <sentence certainty="uncertain" id="S1.10"><*ccue*>*Assuming*</*ccue*> that the 23rd amino acid is also encoded by a stop codon, we systematically predicted proteins that contain stop-codon-encoded amino acids from 191 prokaryotic genomes.</sentence>
- <sentence certainty="uncertain" id="S1.129">A plus sign in a locus <*ccue*>*indicates that*</*ccue*> the genomic coordinates of the iORF can be described by a concatenation of two genes <*ccue*>*or*</*ccue*> regions.</sentence>
- <sentence certainty="uncertain" id="S3.6">We present a novel ensemble learning method, SCOPE, that is based on the <*ccue*>*assumption*</*ccue*> that transcription factor binding sites belong to one of three broad classes of motifs: non-degenerate, degenerate and gapped motifs.</sentence>

tences as certainty or uncertainty.

As a generative model, Naive Bayes simply relaxes the dependency (conditioned on the predicted category) of words amongst one another. The only calculation is to count the words in the training data to estimate the probability that each word associates with the two classes being predicted as certainty or uncertainty plus the prior probability of each class. This assumption saves a lot of memory space and time for building the classifier.

In our preprocessing, we did not conduct stemming or lemmatization on words since we believed that word details (e.g., lower case versus upper-case, plurality, tense, etc.) can be important orthographic features. The Naive Bayes model conducts a sentence-level classification: its output is a class type (i.e., certainty or uncertainty), with no cue phrase identified. We empirically ask whether the bag-of-words approach is complex enough to capture the correlation between word and class on this task.

We also construct another model from sequence-labeling approach, recognizing the hedge-cue information in the sentence can be formalized as a sequence-labeling problem with a specific tag set. For sequence labeling, there is variety of methods such as HMM, CRFs, Averaged Perceptron, etc. [16]. Our second solution is to use an

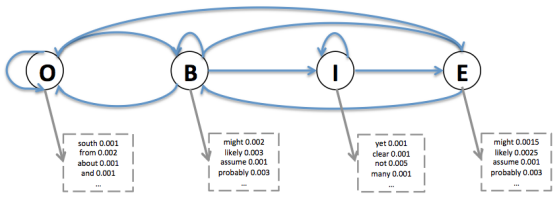

Figure 1: Our HMM topology with a tag set of {O, B, I, E}

HMM with a tag set of {O, B, I, E} for segmentation, where B stands for the beginning of a cue phrase, I the inside of cue phrase, E the end of a cue phrase, and O the outside of a cue phrase. The topology of our HMM is shown in Figure 1, where we consider the first-order dependency between the words. To the best of our knowledge, an HMM has not been applied to the CoNLL 2010 shared task, and most methods of sequence labeling are CRFs. Another goal we have is that we are trying to see whether a less strongly expressive model like HMMs (as compared to CRFs) can represent the dependency for this task effectively.

3.2.1 *Model definition*

Suppose $X_{(1:N)}$ is the observed word sequence, θ is the parameters (i.e., transition matrix and emission matrix) in HMM model, the notation $Z_{(1:N)}$ is a state sequence aligned with the observation word sequence $X_{(1:N)}$, the problem can be formalized to find the most likely state sequence to the corresponding observation, that is, $MAX_{Z_{(1:N)}} p(Z_{(1:N)}|X_{(1:N)}, \theta)$. In our setting, the average length of sentence is 35 words and the maximal is 110 words. We use the Viterbi Algorithm [5] to find the state sequence that maximizes the probability of aligning the tags with the observation based on the parameters. But we need good parameters θ for the model first.

3.2.2 *Parameter estimation*

Data sparsity and high dimensionality are two problems for statistical language processing. In many supervised-style NLP systems, the feature space includes words, while the vocabularies can be extremely large, which leads to a high number of parameters. To make matters worse, many words will appear only a few times, and a large number of words do not appear in the training set. Performance can be degraded on such "out of vocabulary" words. Another phenomenon that degrades performance occurs when the domain of the test set differs from the domain of the training set, in part because the test set includes more words that do not appear or only appear few times in the training set. Thus, good parameter estimation is critical to derive an accurate model.

In our setting, we need to estimate the transition and emission probabilities for the HMM. We tested two strategies in our experiments. One is Maximum Likelihood Estimation (MLE), to find the parameters θ by maximizing the likelihood of the observed data; we combine MLE and the Viterbi algorithm, to form an iterative MLE that co-trains on the labeled data and unlabeled data. Initially, we get the parameters from labeled training examples, and use the model to predict the labels for the unlabeled test set, then we use the whole dataset (including training examples and these predicted test examples) to re-estimate the parameters. In this fashion we obtain an iterative maximum likelihood estimation. Note that we use the test examples in the subsequent iterative runs, but we did not disclose the actual labels of test examples to the training process (which is called *transductive* training), and we use the predicted labels for these test examples from the former iteration. The

Table 2: Results of baseline for sentence-level classification using Naive Bayes classifier (Genre: Wiki – Wikipedia, Bio – biomedical; Type: C – Closed-Domain, X – Cross-Domain).

Genre	P	R	F	Type
Wiki	32.70	80.75	**46.56**	**C**
Wiki	33.05	77.57	46.35	**X**
Bio	31.33	87.85	46.19	**C**
Bio	41.74	51.20	**51.20**	**X**

risk is that there might bring some cumulative errors from previous iterations, so we need to choose a good point to stop the iteration that is done by fitting a development (i.e., "tuning") dataset.

Our other approach is to use the Expectation Maximization formalism by EM [3], or Baum-Welch algorithm [1], or Forward-Backward algorithm to get good parameters for the HMM, because the limited training examples and the sparsity in the language might yield a biased transition matrix and emission matrix (e.g., by underfitting or overfitting). To estimate the parameters by EM, we start from some initial parameters, which can be drawn from those training labeled data only, then iteratively execute the E-step and M-step until convergence. Since EM is not guaranteed to find the global optimal, we used the strategy of multiple restarts by changing the initial parameters for EM algorithm to get a set of local optimal, and chosen the maximal one.

4. EXPERIMENTS AND ANALYSIS

In our experiments, we evaluate the performance of the system under several different configurations. One is whether models are allowed to exploit different kinds of annotated resources: *closed-domain* versus *cross-domain* training. Closed-domain training is where the training set and test set are from the same domain, e.g., Wikipedia training data for a Wikipedia test set. Cross-domain training means that the data provided for the task comes from both domains, e.g., a union of Wikipedia and biomedical training data for either Wikipedia or biomedical test dataset[2].

We ran the evaluation with the standard evaluation scripts provided by CoNLL-2010's Shared Task 1, which used the metrics of precision, recall and F-measure. For this task, not only did the dataset creators provide evaluation on sentence level, but also they provided in-sentence cue annotation evaluation. In both evaluations, we employed the F-measure (i.e., the harmonic mean of precision and recall) as the chief measure metric.

In the bag-of-words solution using the Naive Bayes classifier, we just evaluate its sentence-level classification performance; with no cue annotation that cannot be easily done by using a bag-of-words model to tag the cue phrase at token level. Also, cue annotation is optional in the shared task 1. For the HMM, we evaluate both its sentence-level performance and in-sentence cue annotation.

4.1 Results

4.1.1 Baseline Results using Naive Bayes Classifier

Table 2 gives the best performance results of the baseline for both closed-domain and cross-domain training on the two datasets.

[2]There are two scenarios in CoNLL 2010 shared task 1 for cross-domain training. One uses the union of the two datasets as the training set and tests on one domain, and the other uses the dataset from one domain as training set and test on a different domain. Here we consider the first case

Table 3: Smoothing for sparsity in parameter estimation based on closed-domain training using Naive Bayes classifier.

Genre	P	R	F	Smooth factor
Wiki	38.71	34.83	36.66	(1, 1)
Wiki	34.13	53.27	41.60	(10, 1)
Wiki	31.85	83.44	46.10	(10, 6)
Wiki	32.70	80.75	**46.56**	(11, 5)
Wiki	33.43	76.14	46.46	(11, 4)
Wiki	33.67	70.90	45.66	(12, 3)
Bio	31.33	87.85	**46.19**	(1, 1)
Bio	27.94	90.89	42.74	(1, 2)
Bio	30.58	89.11	45.54	(2, 1)
Bio	30.04	90.38	45.09	(3, 1)

Table 4: Smoothing for sparsity in parameter estimation based on cross-domain training using Naive Bayes classifier.

Genre	P	R	F	Smooth factor
Wiki	34.30	70.90	46.23	(1, 1)
Wiki	31.29	78.25	44.71	(1, 2)
Wiki	33.64	73.50	46.16	(2, 1)
Wiki	33.17	75.74	46.09	(3, 1)
Wiki	33.05	77.57	**46.35**	(4, 1)
Wiki	32.60	78.29	46.03	(5, 1)
Bio	41.74	66.20	**51.20**	(1, 1)
Bio	36.65	70.38	48.20	(1, 2)
Bio	40.24	67.85	50.52	(2, 1)
Bio	38.71	68.99	49.59	(3, 1)

Some interesting findings are as follows:

- For the Naive Bayes model, cross-domain training is helpful on the biomedical dataset. On the Wikipedia dataset, cross-domain training reached comparable performance with that of closed-domain training. Naive Bayes ignores the dependency between words, while cross-domain training introduces the correlation between different domains. In our case, we used both Wikipedia and biomedical training sets as the total training set, the increased volume of training data might contribute to the performance for classification, especially on biomedical dataset.

As we observed, sparsity is a tricky issue to deal with, and smoothing is a relatively easy-to-use method to alleviate sparsity in parameter estimation. In our approach, the smoothing factor is specified by the initial starting (pseudo) count for words and the (pseudo) count for unseen words. Table 3 and 4 contain the results of Naive Bayes under various smoothing factors. From the result of this experiment, we have the following findings:

- Wikipedia data is sensitive to the smoothing technology. It reaches its maximum at a smoothing factor of (11, 5) in closed-domain training; and (4, 1) in cross-domain training.

- Biomedical data is insensitive to the smoothing technology. The best performance for both cross-domain and close-domain training is from default (1, 1).

Table 5: Results of HMM (with iterative MLE) on sentence-level classification (Types: C – Closed-Domain; X – Cross-Domain).

Genre	P	R	F	Iteration	Type
Wiki	69.44	51.25	58.98	1	**C**
Wiki	66.80	58.82	62.56	2	**C**
Wiki	62.11	63.70	**62.90**	3	**C**
Wiki	58.54	66.74	62.37	4	**C**
Wiki	63.72	53.94	58.42	1	**X**
Wiki	60.59	60.97	60.78	2	**X**
Wiki	58.06	66.07	61.81	3	**X**
Wiki	56.84	69.96	**62.72**	4	**X**
Wiki	55.26	72.16	62.59	5	**X**
Bio	75.06	80.76	77.81	1	**C**
Bio	72.80	83.67	**77.86**	2	**C**
Bio	68.20	87.98	76.84	3	**C**
Bio	64.21	66.08	**65.13**	1	**X**

- One interpretation of such difference is that the biomedical context has a relatively more concentrated vocabulary while the Wikipedia text has a more diversified thus more sparse vocabulary. Smoothing in machine learning is a prior to encode some background domain knowledge. Therefore, if our training set is quite large or more concentrated, the variation from smoothing will affect little to the result.

4.1.2 Results of HMM-Based Sentence-level Classification

Table 5 contains the results for sentence-level classification based on HMM under iterative MLE. We obtained the following findings:

- This HMM (with MLE) on Wikipedia dataset using closed-domain training achieves superior performance of 62.90% F score, while the highest performance on CoNLL-2010 submissions from closed-domain training is 60.2%, which is achieved by a SVM classifier using unigrams. HMM is basically a sequence-labeling approach that considers the relation between words, their orders, etc., while the SVM based on bag-of-words does not capture such relations. This implies the HMM algorithm is effective by capturing word dependencies.

- In our cross-domain training on Wikipedia dataset, we achieve a high F score of 62.72%, which is better than any of the submissions in CoNLL-2010. The best performance on cross-domain evaluation is obtained by an Averaged Perceptron with 58.7% F score. We also note that an CRFs model [21] can achieve 55.0% F score, which is much lower than ours on Task 1, but its in-sentence cue annotation (36.5%) is higher ours (18.13%) (see Table 8). One interpretation is that it used the dictionary for cue detection, and in CoNLL 2010 shared tasks, it demonstrated that dictionary vocabulary helps a lot in cue detection and shared task 2.

- We disagree with one statement made by CoNLL-2010 shared task report [4], which summarized that Wikipedia articles have a diverse nature based on the observation that a bag-of-words model achieved the best result among their submissions on the Task 1 Wikipedia dataset. In our experiment, we demonstrated that sequence labeling could achieve better performance on the Wikipedia dataset than that on the biomedical dataset, which reflects that the cue words can still be useful. This is further demonstrated by using the HMM with EM.

- The performance of the HMM (with MLE) on the biomedical dataset is at a state-of-the-art level, though not a top performance in the submissions of CoNLL-2010 Shared Task 1. One possible reason is the parameters derived by MLE are still not optimal to generalize on the biomedical test dataset. Another reason might be that our HMM might not be accurate enough to capture the dependency in the biomedical context for this task; the biomedical context has a more complex nature than that of Wikipedia articles.

- We observed that for the HMM (with MLE) the performance of cross-domain training is not better than that of the corresponding closed-domain training. In contrast, for Naive Bayes the results of cross-domain training are better than closed-domain training. This implies that the task calls for a sophisticated model to express the dependency among words, even though the cross-domain training can provide more training examples for weak model (e.g., Naive Bayes).

Table 6: Results of HMM (with EM) for sentence-level classification based on closed-domain training.

Genre	P	R	F
Wiki	66.91	61.28	**63.97**
Bio	73.32	88.35	**80.15**

Table 6 contains our results for sentence-level classification based on the HMM with EM. In this setting, the initial starting parameters of transition matrix and emission matrix affect the result. In the current EM algorithm, the initial parameters are drawn from the distribution of the labeled training examples. For this experiment, our findings are as follows:

- The performance of the HMM (with EM) on Wikipedia (64.0%) outperforms the best one of HMM (with iterative MLE, 62.9%) since the EM can derive more optimal parameters than the MLE in general, and also exceeds all the submissions in CoNLL-2010 Shared Task 1 on Wikipedia dataset.

- The performance of the HMM (with EM) on biomedical dataset (80.15%) outperforms the best one of HMM (with iterative MLE, 77.86%). However, this result ranks in the middle of the submissions of CoNLL-2010 Shared Task 1 on biomedical dataset whose results range from 30.3% to 86.4%, and the best performance was achieved by CRF. One possible interpretation is that this HMM with a tag set of $\{O, B, I, E\}$ is not expressive enough to capture the dependency of words in the biomedical context even though the sophisticated parameter estimation (e.g., MLE or EM) can boost the performance to a decent level. So the complex structure and distant dependency in biomedical text calls for a more complex model (e.g., CRFs).

4.1.3 Results of HMM-based Cue Annotation

Cue annotation is optional in Shared Task 1 of CoNLL-2010. Table 7 and 8 contain the results for hedge-cue annotation based on

Table 7: Results of HMM (with iterative MLE) for in-sentence cue annotation based on closed-domain training.

Genre	P	R	F	Iteration
Wiki	22.22	11.58	**15.23**	1
Wiki	17.97	12.76	14.92	2
Wiki	15.88	13.55	14.62	3
Bio	39.17	38.68	**38.92**	1
Bio	25.25	27.03	26.11	2

Table 8: Results of HMM (with EM) for in-sentence cue annotation based on closed domain training

Genre	P	R	F
Wiki	20.85	16.04	**18.13**
Bio	23.01	25.60	24.23

HMM (with MLE or EM). The Naive Bayes model is on sentence level, which cannot annotate the hedge cues. The HMM performs token level classification, which associates a class label to each token, and we aggregate the hedge cues from these individual token labels for getting the sentence-level uncertainty labels. Some of our findings are summarized as follows:

- The performance of token-level hedge-cue annotation is not as good as sentence-level classification. This might be caused by the current parameter estimation, which is Expectation Maximization based on using labeled training examples only. The dictionary-based approach was demonstrated good performance for cue annotation in CoNLL-2010 summary report [4]. We did not apply the dictionary approach for detecting cues in sentences. This might be a reason for the poor performance of cue annotation. Actually, most of the systems in CoNLL 2010 shared task used the dictionary for cue detection. In our work, we are using a pure statistical approach. While the cue annotation is hard, it does not affect too much on sentence level detection (see Section 5 for more discussions).
- Table 7 shows that iteration negated the performance of the in-sentence cue annotation in both Wikipedia and biomedical datasets. One possible interpretation is that the errors (i.e., false positives and false negatives) from former iterations will be accumulated into next iteration, which will hurt the performance for future iteration in cue annotation (see Section 5 for more discussions).

5. DISCUSSION

In our bag-of-words solution, cross-domain training can deliver comparable or even better performance than that of closed-domain training. This can reflect that the different testbeds share some correlation or simple patterns. In such a simple model, this kind of correlation might be able to boost the classification to a certain degree. However, in a more sophisticated model (e.g. HMM), the cross-domain training may bring noisy data to bias the parameter estimation. Under this circumstance, the closed-domain training can yield a more accurate model.

In our results, the HMM model outperforms Naive Bayes model by 17.4 percentage points on the Wikipedia dataset and by 29.0 percentage points on the biomedical dataset. This suggests that the Shared Task 1 of CoNLL-2010 calls for more expressive model to count the dependency between words.

Furthermore, we achieved the best performance with a HMM on the Wikipedia dataset, outperforming all of the submissions to the CoNLL-2010 Shared Task 1, including some CRF-based models. However, it is still not confident to conclude that HMM is better than CRFs on Task 1 since it just reflects that the hedge detection at the cue level is not analogous to the sentence-level classification. Our HMM model achieved the best performance on sentence-level, but it was worse on the cue level. Additionally, by the comparison of the same model on the biomedical dataset, the results only rank in the middle in all the submissions of CoNLL-2010. This indicates that there exists some significant difference between the biomedical and Wikipedia contexts. Biomedical scientific contexts (including abstracts and full articles) might have more complex structure and distant dependency among words in sentence, which requires a more expressive model (e.g., CRFs) to capture such relations while HMM only captures the dependency between neighboring words.

Together with Table 7 and 8, we explore more causes regarding why hedge detection at the cue level is not analogous to sentence-level classification. Some reasons might be: sentences with more than one cues are tagged as uncertainty even if only one hedge cue has been identified, which will lead to a lower recall at cue-level annotation evaluation, but can also yield the correct result at sentence-level classification. False positives at cue level can also lead to the correct classification result at the sentence level. This might account for why the F score of cue annotation is lower while sentence-level classification is quite good. To some extent, this also makes sense that iterative re-estimation together with unlabeled data can lead to a better performance (in terms of higher recall and lower precision) in sentence-level classification, but undermine the performance (with a much lower precision) of cue level annotation.

In our HMM model (with iterative MLE), we demonstrated that by combining the unlabeled data with labeled training instances, we can obtain an accurate model on both datasets on sentence-level classification. But in cue annotation level, it is harmful to accurately label the cue token. We are not sure why this is the case at this moment.

In the Naive Bayes model, we observed that data sparsity is a severe problem. We varied the smoothing factor to boost the performance from 36.7% to 46.6% on the Wikipedia dataset. In the experiment, we did not commit much effort on solving the sparsity in language model. But we believe it should be a good direction to continue to estimate the distribution in language since sparsity connects well to the parameter estimation for the model.

6. CONCLUSION AND FUTURE WORK

We addressed detecting sentences containing uncertainty and labeling the cue information in the sentence. Firstly, we provided a bag-of-words model – Naive Bayes – and experimented with a smoothing technique to improve parameter estimation. Secondly, we investigated a sequence labeling model – an HMM – and experimented with different parameter estimation strategies for the model using Maximum Likelihood Estimation and Expectation Maximization. In our experiments, the HMM with EM is the best approach on both the Wikipedia and biomedical datasets.

Compared to the benchmark submissions of CoNLL-2010 Shared Task 1, both our HMM with MLE and EM outperform all the submissions on the Wikipedia dataset, while the same model ranks in the middle of all the submissions on the biomedical dataset. We conclude that capturing the dependency between words yields a more accurate model (e.g. HMM in our experiment), which sug-

gests the use of a more expressive model (e.g., CRFs) that can capture the distant dependency and complex structure in the biomedical text. Further, we experimented with closed-domain and cross-domain trainings. For the Naive Bayes model, cross-domain training worked better, while in a more expressive HMM approach, the best result is obtained from closed-domain training.

For the future, we will apply the uncertainty detection approaches for applications such as information extraction, knowledge-base construction, biomedical literature mining and question answering.

7. ACKNOWLEDGEMENTS

JWS was supported by the DARPA DEFT program under the Air Force Research Laboratory (AFRL) prime contract no. FA8750-13-2- 0039. The opinions in this paper are not necessarily those of DARPA, AFRL, or the US Government.

8. REFERENCES

[1] P. M. Baggenstoss. A modified baum-welch algorithm for hidden markov models with multiple observation spaces. *IEEE Transactions on Speech and Audio Processing*, 9(4):411–416, 2001.

[2] D. Clausen. Hedgehunter: A system for hedge detection and uncertainty classification. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 120–125, 2010.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of The Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[4] R. Farkas, V. Vincze, G. Móra, J. Csirik, and G. Szarvas. The conll-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 1–12, 2010.

[5] G. D. Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61:268 – 278, 1973.

[6] V. Ganter and M. Strube. Finding hedges by chasing weasels: Hedge detection using wikipedia tags and shallow linguistic features. In *Proceedings of the ACL-IJCNLP 2009*, pages 173–176. Association for Computational Linguistics, 2009.

[7] M. Georgescul. A hedgehop over a max-margin framework using hedge cues. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task*, pages 26–31. Association for Computational Linguistics, 2010.

[8] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an em approach. In *Advances in Neural Information Processing Systems 6*, pages 120–127, 1994.

[9] F. Huang and A. Yates. Distributional representations for handling sparsity in supervised sequence-labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, pages 495–503, 2009.

[10] F. Ji, X. Qiu, and X. Huang. Detecting hedge cues and their scopes with average perceptron. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 32–39, 2010.

[11] M. Light, X. Y. Qiu, and P. Srinivasan. The language of bioscience: Facts, speculations, and statements in between. In *Proceedings of BioLink 2004 workshop on linking biological literature, ontologies and databases: tools for users*, pages 17–24, 2004.

[12] E. Marsi and F. v. Rooden. Expressing uncertainty with a talking head in a multimodal question-answering system. In *Proceedings of Workshop Multimodal Output Generation*, pages 105–116, 2007.

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

[14] B. Medlock and T. Briscoe. 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, 2007.

[15] R. Morante, V. Van Asch, and W. Daelemans. Memory-based resolution of in-sentence scopes of hedge cues. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 40–47, 2010.

[16] N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th International Conference on Machine Learning*, pages 681–688, 2007.

[17] R. Saurí and J. Pustejovsky. Factbank: A corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268, 2009.

[18] G. Szarvas. Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics*, 2008.

[19] G. Szarvas, V. Vincze, R. Farkas, G. Móra, and I. Gurevych. Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367, 2012.

[20] O. Täckström, S. Velupillai, M. Hassel, G. Eriksson, H. Dalianis, and J. Karlgren. Uncertainty detection as approximate max-margin sequence labelling. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 84–91, 2010.

[21] B. Tang, X. Wang, X. Wang, B. Yuan, and S. Fan. A cascade method for detecting hedges and their scope in natural language text. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task*, pages 13–17. Association for Computational Linguistics, 2010.

[22] E. Tjong and K. Sang. A baseline approach for detecting sentences containing uncertainty. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 148–150, 2010.

[23] E. Velldal. Predicting speculation: A simple disambiguation approach to hedge detection in biomedical literature. *Journal of Biomedical Semantics*, 2:1–14, 2011.

[24] V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, 9(Suppl 11):S9, 2008.

[25] A. Vlachos and M. Craven. Detecting speculative language using syntactic dependencies and logistic regression. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 18–25, 2010.

[26] Z. Wei, J. Chen, W. Gao, B. Li, L. Zhou, Y. He, and K.-F. Wong. An empirical study on uncertainty identification in

social media context. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*, pages 58–62, 2013.

[27] Q. Zhao, C. Sun, B. Liu, and Y. Cheng. Learning to detect hedges and their scope using crf. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, pages 100–105, 2010.