Weakly Supervised Learning for Hedge Classification in Scientific Literature

Ben Medlock
Computer Laboratory
University of Cambridge
Cambridge, CB3 OFD
benmedlock@cantab.net

Ted Briscoe
Computer Laboratory
University of Cambridge
Cambridge, CB3 OFD
ejb@cl.cam.ac.uk

Abstract

We investigate automatic classification of speculative language ('hedging'), in biomedical text using weakly supervised machine learning. Our contributions include a precise description of the task with annotation guidelines, analysis and discussion, a probabilistic weakly supervised learning model, and experimental evaluation of the methods presented. We show that hedge classification is feasible using weakly supervised ML, and point toward avenues for future research.

1 Introduction

The automatic processing of scientific papers using NLP and machine learning (ML) techniques is an increasingly important aspect of technical informatics. In the quest for a deeper machine-driven 'understanding' of the mass of scientific literature, a frequently occuring linguistic phenomenon that must be accounted for is the use of *hedging* to denote propositions of a speculative nature. Consider the following:

1. *Our results prove that XfK89 inhibits Felin-9.*
2. *Our results suggest that XfK89 might inhibit Felin-9.*

The second example contains a hedge, signaled by the use of *suggest* and *might*, which renders the proposition *inhibit(XfK89→Felin-9)* speculative. Such analysis would be useful in various applications; for instance, consider a system designed to identify and extract interactions between genetic entities in the biomedical domain. Case 1 above provides clear textual evidence of such an interaction and justifies extraction of *inhibit(XfK89→Felin-9)*, whereas case 2 provides only weak evidence for such an interaction.

Hedging occurs across the entire spectrum of scientific literature, though it is particularly common in the experimental natural sciences. In this study we consider the problem of learning to automatically classify sentences containing instances of hedging, given only a very limited amount of annotator-labelled 'seed' data. This falls within the *weakly supervised* ML framework, for which a range of techniques have been previously explored. The contributions of our work are as follows:

1. We provide a clear description of the problem of hedge classification and offer an improved and expanded set of annotation guidelines, which as we demonstrate experimentally are sufficient to induce a high level of agreement between independent annotators.
2. We discuss the specificities of hedge classification as a weakly supervised ML task.
3. We derive a probabilistic weakly supervised learning model and use it to motivate our approach.
4. We analyze our learning model experimentally and report promising results for the task on a new publicly-available dataset.[1]

2 Related Work

2.1 Hedge Classification

While there is a certain amount of literature within the linguistics community on the use of hedging in

[1] available from www.cl.cam.ac.uk/∼bwm23/

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 992–999,
Prague, Czech Republic, June 2007.

scientific text, eg. (Hyland, 1994), there is little of direct relevance to the task of classifying speculative language from an NLP/ML perspective.

The most clearly relevant study is Light et al. (2004) where the focus is on introducing the problem, exploring annotation issues and outlining potential applications rather than on the specificities of the ML approach, though they do present some results using a manually crafted substring matching classifier and a supervised SVM on a collection of *Medline* abstracts. We will draw on this work throughout our presentation of the task.

Hedging is sometimes classed under the umbrella concept of *subjectivity*, which covers a variety of linguistic phenomena used to express differing forms of authorial opinion (Wiebe et al., 2004). Riloff et al. (2003) explore bootstrapping techniques to identify subjective nouns and subsequently classify subjective vs. objective sentences in newswire text. Their work bears some relation to ours; however, our domains of interest differ (newswire vs. scientific text) and they do not address the problem of hedge classification directly.

2.2 Weakly Supervised Learning

Recent years have witnessed a significant growth of research into weakly supervised ML techniques for NLP applications. Different approaches are often characterised as either *multi-* or *single-view*, where the former generate multiple redundant (or semi-redundant) 'views' of a data sample and perform mutual bootstrapping. This idea was formalised by Blum and Mitchell (1998) in their presentation of *co-training*. Co-training has also been used for named entity recognition (NER) (Collins and Singer, 1999), coreference resolution (Ng and Cardie, 2003), text categorization (Nigam and Ghani, 2000) and improving gene name data (Wellner, 2005).

Conversely, single-view learning models operate without an explicit partition of the feature space. Perhaps the most well known of such approaches is *expectation maximization* (EM), used by Nigam et al. (2000) for text categorization and by Ng and Cardie (2003) in combination with a meta-level feature selection procedure. *Self-training* is an alternative single-view algorithm in which a labelled pool is incrementally enlarged with unlabelled samples for which the learner is most confident. Early work by Yarowsky (1995) falls within this framework. Banko and Brill (2001) use 'bagging' and agreement to measure confidence on unlabelled samples, and more recently McClosky et al. (2006) use self-training for improving parse reranking.

Other relevant recent work includes (Zhang, 2004), in which random feature projection and a committee of SVM classifiers is used in a hybrid co/self-training strategy for weakly supervised relation classification and (Chen et al., 2006) where a graph based algorithm called *label propagation* is employed to perform weakly supervised relation extraction.

3 The Hedge Classification Task

Given a collection of sentences, $\mathcal{S}$, the task is to label each sentence as either speculative or non-speculative (*spec* or *nspec* henceforth). Specifically, $\mathcal{S}$ is to be partitioned into two disjoint sets, one representing sentences that contain some form of hedging, and the other representing those that do not.

To further elucidate the nature of the task and improve annotation consistency, we have developed a new set of guidelines, building on the work of Light et al. (2004). As noted by Light et al., speculative assertions are to be identified on the basis of judgements about the author's intended meaning, rather than on the presence of certain designated hedge terms.

We begin with the hedge definition given by Light et al. (item 1) and introduce a set of further guidelines to help elucidate various 'grey areas' and tighten the task specification. These were developed after initial annotation by the authors, and through discussion with colleagues. Further examples are given in online Appendix A[2].

The following are considered hedge instances:

1. An assertion relating to a result that does not necessarily follow from work presented, but could be extrapolated from it (Light et al.).
2. Relay of hedge made in previous work.

   Dl and Ser have been proposed to act redundantly in the sensory bristle lineage.

3. Statement of knowledge paucity.

[2]available from www.cl.cam.ac.uk/~bwm23/

How endocytosis of Dl leads to the activation of N remains to be elucidated.

4. Speculative question.

   A second important question is whether the roX genes have the same, overlapping or complementing functions.

5. Statement of speculative hypothesis.

   To test whether the reported sea urchin sequences represent a true RAG1-like match, we repeated the BLASTP search against all GenBank proteins.

6. Anaphoric hedge reference.

   This hypothesis is supported by our finding that both pupariation rate and survival are affected by EL9.

The following are not considered hedge instances:

1. Indication of experimentally observed non-universal behaviour.

   proteins with single BIR domains can also have functions in cell cycle regulation and cytokinesis.

2. Confident assertion based on external work.

   Two distinct E3 ubiquitin ligases have been shown to regulate Dl signaling in Drosophila melanogaster.

3. Statement of existence of proposed alternatives.

   Different models have been proposed to explain how endocytosis of the ligand, which removes the ligand from the cell surface, results in N receptor activation.

4. Experimentally-supported confirmation of previous speculation.

   Here we show that the hemocytes are the main regulator of adenosine in the Drosophila larva, as was speculated previously for mammals.

5. Negation of previous hedge.

   Although the adgf-a mutation leads to larval or pupal death, we have shown that this is not due to the adenosine or deoxyadenosine simply blocking cellular proliferation or survival, as the experiments in vitro would suggest.

4 Data

We used an archive of 5579 full-text papers from the functional genomics literature relating to Drosophila melanogaster (the fruit fly). The papers were converted to XML and linguistically processed using the RASP toolkit[3]. We annotated six of the papers to form a test set with a total of 380 *spec* sentences and 1157 *nspec* sentences, and randomly selected 300,000 sentences from the remaining papers as training data for the weakly supervised learner. To ensure selection of complete sentences rather than headings, captions etc., unlabelled samples were chosen under the constraints that they must be at least 10 words in length and contain a main verb.

[3]www.informatics.susx.ac.uk/research/nlp/rasp

	F_1^{rel}	κ
Original	0.8293	0.9336
Corrected	0.9652	0.9848

Table 1: Agreement Scores

5 Annotation and Agreement

Two separate annotators were commissioned to label the sentences in the test set, firstly one of the authors and secondly a domain expert with no prior input into the guideline development process. The two annotators labelled the data independently using the guidelines outlined in section 3. Relative F_1 (F_1^{rel}) and *Cohen's Kappa* (κ) were then used to quantify the level of agreement. For brevity we refer the reader to (Artstein and Poesio, 2005) and (Hripcsak and Rothschild, 2004) for formulation and discussion of κ and F_1^{rel} respectively.

The two metrics are based on different assumptions about the nature of the annotation task. F_1^{rel} is founded on the premise that the task is to recognise and label *spec* sentences from within a background population, and does not explicitly model agreement on *nspec* instances. It ranges from 0 (no agreement) to 1 (no disagreement). Conversely, κ gives explicit credit for agreement on both *spec* and *nspec* instances. The observed agreement is then corrected for 'chance agreement', yielding a metric that ranges between -1 and 1. Given our definition of hedge classification and assessing the manner in which the annotation was carried out, we suggest that the founding assumption of F_1^{rel} fits the nature of the task better than that of κ.

Following initial agreement calculation, the instances of disagreement were examined. It turned out that the large majority of cases of disagreement were due to negligence on behalf of one or other of the annotators (i.e. cases of clear hedging that were missed), and that the cases of genuine disagreement were actually quite rare. New labelings were then created with the negligent disagreements corrected, resulting in significantly higher agreement scores. Values for the original and negligence-corrected labelings are reported in Table 1.

Annotator conferral violates the fundamental assumption of annotator independence, and so the latter agreement scores do not represent the true level of agreement; however, it is reasonable to conclude that the actual agreement is approximately lower bounded by the initial values and upper bounded by the latter values. In fact even the lower bound is well within the range usually accepted as representing 'good' agreement, and thus we are confident in accepting human labeling as a gold-standard for the hedge classification task. For our experiments, we use the labeling of the genetics expert, corrected for negligent instances.

6 Discussion

In this study we use single terms as features, based on the intuition that many hedge cues are single terms (*suggest*, *likely* etc.) and due to the success of 'bag of words' representations in many classification tasks to date. Investigating more complex sample representation strategies is an avenue for future research.

There are a number of factors that make our formulation of hedge classification both interesting and challenging from a weakly supervised learning perspective. Firstly, due to the relative sparsity of hedge cues, most samples contain large numbers of irrelevant features. This is in contrast to much previous work on weakly supervised learning, where for instance in the case of text categorization (Blum and Mitchell, 1998; Nigam et al., 2000) almost all content terms are to some degree relevant, and irrelevant terms can often be filtered out (e.g. stopword removal). In the same vein, for the case of entity/relation extraction and classification (Collins and Singer, 1999; Zhang, 2004; Chen et al., 2006) the context of the entity or entities in consideration provides a highly relevant feature space.

Another interesting factor in our formulation of hedge classification is that the *nspec* class is defined on the basis of the *absence* of hedge cues, rendering it hard to model directly. This characteristic is also problematic in terms of selecting a reliable set of *nspec* seed sentences, as by definition at the beginning of the learning cycle the learner has little knowledge about what a hedge looks like. This problem is addressed in section 10.3.

In this study we develop a learning model based around the concept of iteratively predicting labels for unlabelled training samples, the basic paradigm for both co-training and self-training. However we generalise by framing the task in terms of the acquisition of labelled training data, from which a supervised classifier can subsequently be learned.

7 A Probabilistic Model for Training Data Acquisition

In this section, we derive a simple probabilistic model for acquiring training data for a given learning task, and use it to motivate our approach to weakly supervised hedge classification.

Given:

- sample space $\mathcal{X}$
- set of target concept classes $\mathcal{Y} = \{y_1 \dots y_N\}$
- target function $Y : \mathcal{X} \rightarrow \mathcal{Y}$
- set of seed samples for each class $\mathcal{S}_1 \dots \mathcal{S}_N$ where $\mathcal{S}_i \subset \mathcal{X}$ and $\forall \mathbf{x} \in \mathcal{S}_i[Y(\mathbf{x})=y_i]$
- set of unlabelled samples $\mathcal{U} = \{\mathbf{x}_1 \dots \mathbf{x}_K\}$

Aim: *Infer a set of training samples $\mathcal{T}_i$ for each concept class y_i such that* $\forall \mathbf{x} \in \mathcal{T}_i[Y(\mathbf{x}) = y_i]$

Now, it follows that $\forall \mathbf{x} \in \mathcal{T}_i[Y(\mathbf{x})=y_i]$ is satisfied in the case that $\forall \mathbf{x} \in \mathcal{T}_i[P(y_i|x)=1]$, which leads to a model in which $\mathcal{T}_i$ is initialised to $\mathcal{S}_i$ and then iteratively augmented with the unlabelled sample(s) for which the posterior probability of class membership is maximal. Formally:

At each iteration:

$$\mathcal{T}_i \leftarrow \mathbf{x}_j (\in \mathcal{U}) \quad \text{where } j = \arg\max_j [P(y_i|\mathbf{x}_j)] \tag{1}$$

Expansion with Bayes' Rule yields:

$$\begin{aligned} & \arg\max_j \left[P(y_i|\mathbf{x}_j)\right] \\ = \quad & \arg\max_j \left[\frac{P(\mathbf{x}_j|y_i) \cdot P(y_i)}{P(\mathbf{x}_j)}\right] \end{aligned} \tag{2}$$

An interesting observation is the importance of the sample prior $P(\mathbf{x}_j)$ in the denominator, often ignored for classification purposes because of its invariance to class. We can expand further by marginalising over the classes in the denominator in expression 2, yielding:

$$\arg\max_j \left[\frac{P(\mathbf{x}_j|y_i) \cdot P(y_i)}{\sum_{n=1}^{N} P(y_n)P(\mathbf{x}_j|y_n)} \right] \quad (3)$$

so we are left with the class priors and class-conditional likelihoods, which can usually be estimated directly from the data, at least under limited dependence assumptions. The class priors can be estimated based on the relative distribution sizes derived from the current training sets:

$$P(y_i) = \frac{|\mathcal{T}_i|}{\sum_k |\mathcal{T}_k|} \quad (4)$$

where $|\mathcal{S}|$ is the number of samples in training set $\mathcal{S}$.

If we assume feature independence, which as we will see for our task is not as gross an approximation as it may at first seem, we can simplify the class-conditional likelihood in the well known manner:

$$P(\mathbf{x}_j|y_i) = \prod_k P(x_{jk}|y_i) \quad (5)$$

and then estimate the likelihood for each feature:

$$P(x_k|y_i) = \frac{\alpha P(y_i) + f(x_k, \mathcal{T}_i)}{\alpha P(y_i) + |\mathcal{T}_i|} \quad (6)$$

where $f(x, \mathcal{S})$ is the number of samples in training set $\mathcal{S}$ in which feature x is present, and α is a universal smoothing constant, scaled by the class prior. This scaling is motivated by the principle that without knowledge of the true distribution of a particular feature it makes sense to include knowledge of the class distribution in the smoothing mechanism. Smoothing is particularly important in the early stages of the learning process when the amount of training data is severely limited resulting in unreliable frequency estimates.

8 Hedge Classification

We will now consider how to apply this learning model to the hedge classification task. As discussed earlier, the speculative/non-speculative distinction hinges on the presence or absence of a few hedge cues within the sentence. Working on this premise, all features are ranked according to their probability of 'hedge cue-ness':

$$P(spec|x_k) = \frac{P(x_k|spec) \cdot P(spec)}{\sum_{n=1}^{N} P(y_n)P(x_k|y_n)} \quad (7)$$

which can be computed directly using (4) and (6). The m most probable features are then selected from each sentence to compute (5) and the rest are ignored. This has the dual benefit of removing irrelevant features and also reducing dependence between features, as the selected features will often be non-local and thus not too tightly correlated.

Note that this idea differs from traditional feature selection in two important ways:

1. Only features indicative of the *spec* class are retained, or to put it another way, *nspec* class membership is inferred from the absence of strong *spec* features.
2. Feature selection in this context is *not* a preprocessing step; i.e. there is no re-estimation after selection. This has the potentially detrimental side effect of skewing the posterior estimates in favour of the *spec* class, but is admissible for the purposes of ranking and classification by posterior thresholding (see next section).

9 Classification

The weakly supervised learner returns a labelled data set for each class, from which a classifier can be trained. We can easily derive a classifier using the estimates from our learning model by:

$$\mathbf{x}_j \rightarrow spec \quad \text{if} \quad P(spec|\mathbf{x}_j) > \sigma \quad (8)$$

where σ is an arbitrary threshold used to control the precision/recall balance. For comparison purposes, we also use Joachims' SVM^{light} (Joachims, 1999).

10 Experimental Evaluation

10.1 Method

To examine the practical efficacy of the learning and classification models we have presented, we use the following experimental method:

1. Generate seed training data: $\mathcal{S}_{spec}$ and $\mathcal{S}_{nspec}$
2. Initialise: $\mathcal{T}_{spec} \leftarrow \mathcal{S}_{spec}$ and $\mathcal{T}_{nspec} \leftarrow \mathcal{S}_{nspec}$
3. Iterate:
    - Order $\mathcal{U}$ by $P(spec|\mathbf{x}_j)$ (expression 3)
    - $\mathcal{T}_{spec}$ ← most probable batch
    - $\mathcal{T}_{nspec}$ ← least probable batch
    - Train classifier using $\mathcal{T}_{spec}$ and $\mathcal{T}_{nspec}$

Rank	$\alpha = 0$	$\alpha = 1$	$\alpha = 5$	$\alpha = 100$	$\alpha = 500$
1	interactswith	suggest	suggest	suggest	suggest
2	TAFb	likely	likely	likely	likely
3	sexta	may	may	may	may
4	CRYs	might	might	These	These
5	DsRed	seems	seems	results	results
6	Cell-Nonautonomous	suggests	Taken	might	that
7	arva	probably	suggests	observations	be
8	inter-homologue	suggesting	probably	Taken	data
9	Mohanty	possibly	Together	findings	it
10	meld	suggested	suggesting	Our	Our
11	aDNA	Taken	possibly	seems	observations
12	Deer	unlikely	suggested	together	role
13	Borel	Together	findings	Together	most
14	substripe	physiology	observations	role	these
15	Failing	modulated	Given	that	together

Table 2: Features ranked by $P(spec|x_k)$ for varying α

- Compute *spec* recall/precision BEP (break-even point) on the test data

The batch size for each iteration is set to $0.001 * |\mathcal{U}|$. After each learning iteration, we compute the precision/recall BEP for the *spec* class using both classifiers trained on the current labelled data. We use BEP because it helps to mitigate against misleading results due to discrepancies in classification threshold placement. Disadvantageously, BEP does not measure a classifier's performance across the whole of the recall/precision spectrum (as can be obtained, for instance, from receiver-operating characteristic (ROC) curves), but for our purposes it provides a clear, abstracted overview of a classifier's accuracy given a particular training set.

10.2 Parameter Setting

The training and classification models we have presented require the setting of two parameters: the smoothing parameter α and the number of features per sample m. Analysis of the effect of varying α on feature ranking reveals that when $\alpha = 0$, low frequency terms with spurious class correlation dominate and as α increases, high frequency terms become increasingly dominant, eventually smoothing away genuine low-to-mid frequency correlations. This effect is illustrated in Table 2, and from this analysis we chose $\alpha = 5$ as an appropriate level of smoothing. We use $m = 5$ based on the intuition that five is a rough upper bound on the number of hedge cue features likely to occur in any one sentence.

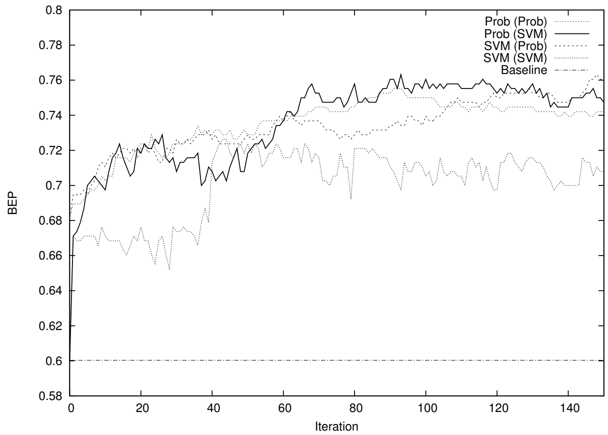
We use the linear kernel for SVM^{light} with the default setting for the regularization parameter C. We construct binary valued, L_2-normalised (unit length) input vectors to represent each sentence, as this resulted in better performance than using frequency-based weights and concords with our presence/absence feature estimates.

10.3 Seed Generation

The learning model we have presented requires a set of seeds for each class. To generate seeds for the *spec* class, we extracted all sentences from $\mathcal{U}$ containing either (or both) of the terms *suggest* or *likely*, as these are very good (though not perfect) hedge cues, yielding 6423 *spec* seeds. Generating seeds for *nspec* is much more difficult, as integrity requires the absence of hedge cues, and this cannot be done automatically. Thus, we used the following procedure to obtain a set of *nspec* seeds:

1. Create initial $\mathcal{S}_{nspec}$ by sampling randomly from $\mathcal{U}$.
2. Manually remove more 'obvious' speculative sentences using pattern matching
3. Iterate:
    - Order $\mathcal{S}_{nspec}$ by $P(spec|\mathbf{x}_j)$ using estimates from $\mathcal{S}_{spec}$ and current $\mathcal{S}_{nspec}$
    - Examine most probable sentences and remove speculative instances

We started with 8830 sentences and after a couple of hours work reduced this down to a (still potentially noisy) *nspec* seed set of 7541 sentences.



Prob (Prob)	denotes our probabilistic learning model and classifier (§9)
Prob (SVM)	denotes probabilistic learning model with SVM classifier
SVM (Prob)	denotes committee-based model (§10.4) with probabilistic classifier
SVM (SVM)	denotes committee-based model with SVM classifier
Baseline	denotes substring matching classifier of (Light et al., 2004)

Figure 1: Learning curves

10.4 Baselines

As a baseline classifier we use the substring matching technique of (Light et al., 2004), which labels a sentence as *spec* if it contains one or more of the following: *suggest*, *potential*, *likely*, *may*, *at least*, *in part*, *possibl*, *further investigation*, *unlikely*, *putative*, *insights*, *point toward*, *promise* and *propose*.

To provide a comparison for our learning model, we implement a more traditional self-training procedure in which at each iteration a committee of five SVMs is trained on randomly generated overlapping subsets of the training data and their cumulative confidence is used to select items for augmenting the labelled training data. For similar work see (Banko and Brill, 2001; Zhang, 2004).

10.5 Results

Figure 1 plots accuracy as a function of the training iteration. After 150 iterations, all of the weakly supervised learning models are significantly more accurate than the baseline according to a binomial sign test ($p < 0.01$), though there is clearly still much room for improvement. The baseline classifier achieves a BEP of 0.60 while both classifiers using our learning model reach approximately 0.76 BEP with little to tell between them. Interestingly, the combination of the SVM committee-based learning model with our classifier (denoted by 'SVM (Prob)'), performs competitively with both of the approaches that use our probabilistic learning model and significantly better than the SVM committee-based learning model with an SVM classifier, 'SVM (SVM)', according to a binomial sign test ($p < 0.01$) after 150 iterations. These results suggest that performance may be enhanced when the learning and classification tasks are carried out by different models. This is an interesting possibility, which we intend to explore further.

An important issue in incremental learning scenarios is identification of the optimum stopping point. Various methods have been investigated to address this problem, such as 'counter-training' (Yangarber, 2003) and committee agreement (Zhang, 2004); how such ideas can be adapted for this task is one of many avenues for future research.

10.6 Error Analysis

Some errors are due to the variety of hedge forms. For example, the learning models were unsuccessful in identifying assertive statements of knowledge paucity, eg: *There is no clear evidence for cytochrome c release during apoptosis in C elegans or Drosophila.* Whether it is possible to learn such examples without additional seed information is an open question. This example also highlights the potential benefit of an enriched sample representation, in this case one which accounts for the negation of the phrase 'clear evidence' which otherwise might suggest a strongly non-speculative assertion.

In many cases hedge classification is challenging even for a human annotator. For instance, distinguishing between a speculative assertion and one relating to a pattern of observed non-universal behaviour is often difficult. The following example was chosen by the learner as a *spec* sentence on the 150th training iteration: *Each component consists of a set of subcomponents that can be localized within a larger distributed neural system.* The sentence does not, in fact, contain a hedge but rather a statement of observed non-universal behaviour. However, an almost identical variant with 'could' instead of 'can' would be a strong speculative candidate. This highlights the similarity between many hedge and non-hedge instances, which makes such cases hard to learn in a weakly supervised manner.

11 Conclusions and Future Work

We have shown that weakly supervised ML is applicable to the problem of hedge classification and that a reasonable level of accuracy can be achieved. The work presented here has application in the wider academic community; in fact a key motivation in this study is to incorporate hedge classification into an interactive system for aiding curators in the construction and population of gene databases. We have presented our initial results on the task using a simple probabilistic model in the hope that this will encourage others to investigate alternative learning models and pursue new techniques for improving accuracy. Our next aim is to explore possibilities of introducing linguistically-motivated knowledge into the sample representation to help the learner identify key hedge-related sentential components, and also to consider hedge classification at the granularity of assertions rather than text sentences.

Acknowledgements

This work was partially supported by the FlySlip project, BBSRC Grant BBS/B/16291, and we thank Nikiforos Karamanis and Ruth Seal for thorough annotation and helpful discussion. The first author is supported by an University of Cambridge Millennium Scholarship.

References

Ron Artstein and Massimo Poesio. 2005. Kappa^3 = alpha (or beta). Technical report, University of Essex Department of Computer Science.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Meeting of the Association for Computational Linguistics*, pages 26–33.

Avrim Blum and Tom Mitchell. 1998. Combining labelled and unlabelled data with co-training. In *Proceedings of COLT' 98*, pages 92–100, New York, NY, USA. ACM Press.

Jinxiu Chen, Donghong Ji, Chew L. Tan, and Zhengyu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of ACL'06*, pages 129–136.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*.

George Hripcsak and Adam Rothschild. 2004. Agreement, the f-measure, and reliability in information retrieval. *J Am Med Inform Assoc.*, 12(3):296–298.

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

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

M. Light, X.Y. Qiu, and P. Srinivasan. 2004. 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, Boston, May 2004*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *HLT-NAACL*.

Vincent Ng and Claire Cardie. 2003. Weakly supervised natural language learning without redundant views. In *Proceedings of NAACL '03*, pages 94–101, Morristown, NJ, USA.

K. Nigam and R. Ghani. 2000. Understanding the behavior of co-training. In *Proceedings of KDD-2000 Workshop on Text Mining*.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Seventh Conference on Natural Language Learning (CoNLL-03). ACL SIGNLL.*, pages 25–32.

Ben Wellner. 2005. Weakly supervised learning methods for improving the quality of gene name normalization data. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases*, pages 1–8, Detroit, June. Association for Computational Linguistics.

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

Roman Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of ACL'03*, pages 343–350, Morristown, NJ, USA.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL'95*, pages 189–196, Morristown, NJ, USA. ACL.

Zhu Zhang. 2004. Weakly-supervised relation classification for information extraction. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 581–588, New York, NY, USA. ACM Press.