Evidence Combination in Biomedical Natural-Language Processing

Marios Skounakis
Dept. of Computer Sciences
Dept. of Biostatistics and Medical Informatics
University of Wisconsin
Madison, WI 53706
marios@cs.wisc.edu

Mark Craven
Dept. of Biostatistics and Medical Informatics
Dept. of Computer Sciences
University of Wisconsin
Madison, WI 53706
craven@biostat.wisc.edu

ABSTRACT

In many natural language tasks, such as information extraction and semantic lexicon building, individual entities and relations of interest may be found in multiple contexts within the corpus. In deciding which putative entities and relations should be extracted, a key problem is how to combine evidence across the multiple occurrences of these entities and relations. We present a novel statistical approach to address this issue, and evaluate it in the context of extracting protein names and protein-protein interactions from MEDLINE abstracts. We experimentally compare our method against a number of intuitive and simpler baselines. Our experimental results suggest that the issue of combining evidence is indeed important in these tasks. Furthermore, we show that our proposed method outperforms the baselines considered in a variety of settings.

Keywords

machine learning, information extraction, text mining

1. INTRODUCTION

There has been much recent interest in applying text-mining methods to the on-line, biomedical literature [5]. These methods have great potential to assist scientists in key tasks such as curating genome databases [19] and annotating high-throughput experiments [16]. One interesting, yet largely unexplored, aspect of mining the biomedical literature is that information of interest usually occurs redundantly. Consider the task of *information extraction* (IE) which involves automatically extracting instances of specified classes, relations or events from text sources. For example, suppose we are interested in extracting information about pairs of proteins that physically interact. In this case, a particular protein-protein interaction might be described in multiple articles, and even in multiple places within each article. The problem that we consider in this paper is how to combine evidence, across different passages of text, for several biomedical text-mining tasks. We present a formal definition of the problem, describe a statistical method for addressing it, and empirically compare our approach against several simpler, ad-hoc methods. Our experiments involve two tasks using a corpus consisting of MEDLINE abstracts [10]. Our experiments indicate that all of the considered methods for combining evidence result in improved extraction accuracy. Furthermore, our experiments show that our approach results in more accurate extractions than the baselines.

To better illustrate the task we are addressing, let us consider the protein-protein interaction case in more detail. Given the text of scientific articles (or just their abstracts), we would like to extract instances of a binary relation that represents the physical interaction of pairs of proteins. For example, given the first or the second sentence shown in Table 1, an accurate IE model would extract the relation instance `protein-interaction(STE7, FUS3)`. For pedagogical simplicity, we assume here that protein names consist of single words. We can then denote a particular candidate extraction by $b_{i,j}$, where w_i and w_j refer to the words naming the proteins in the relation (assuming some unique numbering for the words in the vocabulary). For a given $b_{i,j}$, there might be multiple passages of text in which w_i and w_j occur together. As illustrated in Table 1, some of these co-occurrences might correspond to assertions in text that confirm the relation, whereas others do not support the relation. For example, sentences 1 and 2 in Table 1 are examples of sentences that assert the interaction between `STE7` and `FUS3` (called *positive* occurrences hereafter), whereas sentence 3 does not (*negative* occurrence).

For many applications of text-mining methods to the biomedical literature, we do not need to be especially concerned about the accuracy of our model on isolated passages of text, such as the sentences in Table 1. Instead the primary concern is the corpus-wide accuracy of our model. For example, consider the case in which a model mistakenly does not extract a relation instance from Sentence 2 in Table 1. As long as the model does correctly extract the relation instance from Sentence 1, this mistake is not costly because we will have extracted it from somewhere.

For each occurrence of w_i, w_j, the IE tool will make a *prediction* whether it considers the occurrence to be positive or negative. The IE tool is likely to make mistakes, both false positive predictions (i.e., calling a negative occurrence positive) and false negative predictions (i.e., calling a positive occurrence negative). If we had access to "the truth" about each sentence, then it would be easy to determine if a candidate protein-interaction assertion is true: it suffices to have at least one positive occurrence to call an assertion *positive*. Our goal, since we do not have access to the truth, is to predict how likely it is that an assertion $b_{i,j}$ is positive

	Sentence	Target Extraction
1	From this data, we argue that STE7 is a physiological activator of FUS3.	protein-interaction(STE7, FUS3)
2	Here we report that STE7 is a dual-specificity kinase that modifies FUS3...	protein-interaction(STE7, FUS3)
3	None of the mutations increased the affinity of STE5 for STE11, STE7, or FUS3.	

Table 1: Three sentences that contain occurrences of STE7 and FUS3. We would want our IE model to extract the relation protein-interaction(STE7, FUS3), from Sentences 1 and 2. Since Sentence 3 does not assert that the two proteins interact, we would not want our model to extract a relation instance from it.

Assertion	Pos. Preds	Neg. Preds	Total
$b_{1,2}$	1	2	3
$b_{1,3}$	1	999	1000
$b_{1,4}$	5	5	10

Table 2: A hypothetical case showing three candidate protein-interaction assertions and the number of sentences classified as positive and negative.

given the predictions of the IE tool about the co-occurrences of w_i and w_j.

We can define the task we are addressing as follows.

Given:

- a test corpus to be processed by an information-extraction model,
- estimates of the true-positive and false-positive rates of the model,
- *passage-level* predictions made by the model,

Return: *corpus-level* predictions ranked by confidence.

By *passage-level* predictions here, we mean predictions made on small passages of text, such as the sentences in Table 1. By *corpus-level* predictions we mean the predictions made by aggregating the predictions made across all passages in the corpus. Note that the crux of the task is to rank the corpus-level predictions using information about the IE model and the corpus itself.

Table 2 shows a hypothetical situation in which we have three candidate assertions along with the number of occurrences classified by our model as positive and negative for each assertion. Based on these numbers, we try to decide which assertion is more likely to be positive. By comparing $b_{1,2}$ and $b_{1,3}$ we can conclude that $b_{1,2}$ is more likely to be a true protein interaction than $b_{1,3}$. For both assertions there is a single positive prediction, but for $b_{1,3}$ it is more likely to be a false positive prediction (making some assumptions such as that all predictions are independent) due to its many more total occurrences. In a similar way, we may conclude that $b_{1,4}$ is more likely to be positive than $b_{1,2}$, since it is highly unlikely that *all* five positive predictions are false positives, especially when the total number of occurrences is ten. These examples show that to predict the class of an assertion we must take into account issues such as the number of positive and negative predictions on individual occurrences and the error rates of the IE method.

Our formulation of this problem has the following characteristics. (i) Instances can be grouped into *bags*. In the example in Section 1, an instance is an occurrence of a word pair in a sentence and a bag is the collection of all occurrences of the word pair. (ii) A bag is considered positive if and only if it contains at least one positive instance, and negative otherwise. With respect to the running example, a pair of words represents a protein-interaction assertion if there is at least one sentence in the corpus which states that the proteins do interact. (iii) The classes of the bags correspond to the solutions of the problem. In other words, we are interested in predicting the class of each bag in the corpus. In the protein-interaction task, all bags classified as positive correspond to protein-interaction assertions extracted from the corpus. (iv) We are given a tool that can predict the classes of individual instances. Our goal is to predict the classes of the bags based on the instance predictions.

Many natural language tasks can be mapped to this problem definition. In addition to the protein-interaction task, we also consider here a task that involves constructing a lexicon of protein names given a corpus of biomedical papers. We refer to this as the *protein-lexicon* task. In this task, a bag b_w corresponds to a word w that is a candidate for inclusion in the lexicon, and the instances in a bag represent the occurrences of the word in the corpus. For example all nouns and adjectives (e.g., *Ste11p-binding*) in the corpus might be considered as candidates for inclusion in the lexicon. A positive occurrence (instance) of w is one where the sentence asserts that w is a protein. It suffices to have at least one positive instance of w to conclude that it is a protein.

A number of factors render these tasks challenging. (i) Positive bags may also contain negative instances; thus the number of positive predictions in a bag is not necessarily a good indicator of the likelihood that the bag is positive. (ii) Negative bags may contain multiple instances and as a result there is a high probability for false-positive errors when making predictions for bags, as opposed to individual instances. On the contrary, the fact that positive bags may often contain more than one positive instances decreases the likelihood of false-negative errors.

2. RELATED WORK

For the most part, research in IE has focused on developing methods that operate on relatively short passages of text, such as sentences [1; 3; 8; 12; 14; 17; 18]. These methods treat the corpus as a collection of independent passages. Thus, they are unable to exploit the redundancy inherent in large corpora. The work that we describe here does not constitute a novel IE algorithm. Instead we present a general method for combining evidence that can be used in conjunction with any IE approach.

The problem of evidence combination in information-extraction has been explored by only a few groups. Roth and Yih have developed an approach for IE tasks that uses evidence about potential entities to influence relation extractions, and similarly evidence about potential relations to influence entity extractions [15]. This approach takes advantage of de-

pendencies among entities and relations of interest. Our work is complementary to theirs in that we focus on combining evidence across *multiple contexts* for a single task in isolation. The problem of evidence combination has been touched upon by Riloff's work in semantic lexicon building [13; 14]. Also, Magnini *et al.* [7]. exploit the inherent redundancy of the web the improve the accuracy of question answering, by judging the connection of a candidate answer to the question by the number of web documents in which they co-occur. In contrast to these two efforts, we present a method that is more principled and more general.

Krauthammer *et al.* have proposed a statistical model that characterizes how assertions about molecular interactions entered and are amplified in scientific articles [6]. Their model also includes several parameters that describe how likely an information-extraction system is to extract something about a particular assertion, and how likely it is to be correct about such an extraction. Unlike the work presented herein, however, they have not yet operationalized or empirically evaluated this component of their model.

Also, we note that there is a connection between our problem formulation and that of *multiple-instance* learning tasks [4]. Multiple instance learning focuses on learning from training data that is organized into bags. On the contrary, we focus on making better inferences by grouping instances into bags.

Finally, there is some relationship between our work and *multi-view* learning [2; 9]. As in multi-view learning, our work assumes that there are multiple, independent views that can be used to make a decision about a given candidate extraction. In particular, we can think of the different occurrences of each candidate extraction as corresponding to views since each one occurs in a different context. However, unlike multi-view learning, we do not use different representations for the multiple views and we are not focused on the learning task. Instead, we are interested in using the multiple views at test time in order to get more accurate predictions.

3. BAYESIAN EVIDENCE COMBINATION OF INSTANCE PREDICTIONS

In this section we present an approach, called Bayesian Evidence Combination of Instance Predictions (`BECIP`), for estimating the probability that a bag is positive given the classifier's predictions on the instances of the bag. We assume that for a bag b we are given: (i) the number n_b of instances in the bag (bag size), and (ii) the number k_b of instances for which the classifier predicted the positive class. We also assume that we can estimate: (i) the false positive rate f of the classifier, that is the probability of incorrectly predicting the positive class for a negative instance, (ii) the true positive rate t of the classifier, that is the probability of correctly predicting the positive class for a positive instance, and (iii) the *prior* probability of positive and negative bags given the bag size. Our goal is to compute the probability of bag b being positive. This is equal to the probability $P(m_b > 0|n_b, k_b)$ that the number of positive instances m_b[1] in the bag is larger than zero, since by definition a bag is positive if and only if it contains at least one positive instance.

We make the assumption that the class predicted by the IE tool for an instance $e_i \in b$ is independent of the predictions for the other instances in b. Given f and t and the assumption of independent predictions, we can model the behavior of the classifier on a single instance e as a Bernoulli trial, with "success" being equivalent to predicting the positive class for e. If e is negative, then the probability of success is f. If e is positive, the probability of success is t. The behavior of the classifier on the set of positive instances of a bag b can be modeled with a binomial distribution with probability of success t, denoted as $B(m_b, t)$. Similarly, for the set of negative instances of b we have $B(n_b - m_b, f)$. Refining this model to use a multinomial distribution, with possible outcomes being predictions with varying levels of confidence (i.e., *high confidence positive, medium confidence positive, high confidence negative*, etc.) is straightforward, provided that the classifier can assign confidences to its predictions.

Assuming that there are m_b positive instances in b, the probability of making k_b positive predictions based on our binomial model is:

$$\begin{aligned} P(k_b|m_b, n_b, \theta) &= \sum_{i=0}^{k_b} P(i; B(m_b, t)) P(k_b - i; B(n - m_b, f)) \\ &= \sum_{i=0}^{k_b} \left[\binom{i}{m_b} t^i (1-t)^{(m_b - i)} \times \right. \\ &\quad \left. \binom{k_b - i}{n_b - m_b} f^{(k_b - i)} (1-f)^{(n_b - m_b + i)} \right] \end{aligned}$$

where $P(i; B(m_b, t))$ is the probability of the i of k_b successes (positive predictions) coming from the positive instances and $P(k_b - i; B(n - m_b, f))$ is the probability of the rest of the positive predictions being false positive predictions. θ denotes that this probability estimate is based on the true and false positive rates of the classifier.

Using Bayes' rule we express the posterior probability $P(m_b > 0|n_b, k_b, \theta)$ as:

$$\begin{aligned} P(m_b > 0|n_b, k_b, \theta) &= \frac{P(k_b, m_b > 0, n_b, \theta)}{P(k_b, n_b)} \\ &= \frac{P(k_b|m_b > 0, n_b, \theta) P(m_b > 0, n_b)}{\sum_{i=0}^{n_b} P(k_b, m_b = i, n_b)} \\ &= \frac{\sum_{j=1}^{n_b} P(k_b|m_b = j, n_b, \theta) P(m_b = j|n_b)}{\sum_{i=0}^{n_b} P(k_b|m_b = i, n_b, \theta) P(m_b = i|n_b)} \quad (1) \end{aligned}$$

$P(m_b > 0|n_b, k_b, \theta)$ is our estimate for $P(m_b > 0|n_b, k_b)$ based on the abstraction of the binomial to model the classifier's behavior. We can view this probability estimate as a confidence $C_+(b)$ that bag b is positive. Depending on an application's recall and precision requirements, we can use a threshold d to assign the positive class only to bags for which $C_+(b) > d$.

The assumption that the classifier's predictions on the instances of a bag are independent is often violated. For instances that have lexical and grammatical similarities (e.g., a word that appears in two very similar contexts) the classifier's decisions are likely to be correlated. The extent to which the independence assumption is violated can play a key role in the accuracy of `BECIP`'s estimates. This issue must be kept in mind when applying `BECIP`.

[1] We emphasize the difference between m_b, the number of positive instances in b, and k_b, the number of positive *predictions* made by the classifier for b. The latter includes the false-positive and false-negative errors that the classifier can make and will often have a different value than m_b.

3.1 Estimating the False Positive and True Positive Rates

The false positive f and true positive t rates of the classifier can be estimated using a held-out portion H of the labeled corpus, D. The classifier is trained on the rest of the labeled set $D - H$ and then tested on H, for which the class labels are known. With sufficiently large H accurate estimates for f and t can be acquired. If the classifier is hand-constructed then all of D can be used to estimate t and f.

3.2 Estimating the Priors

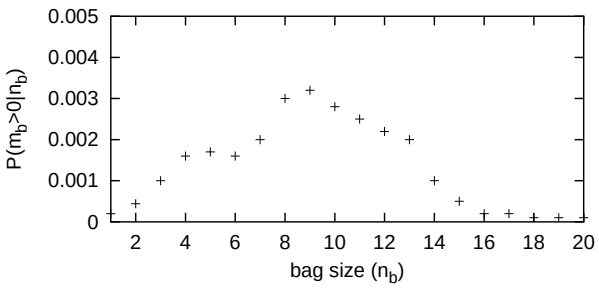
The use of Equation 1 requires estimating $P(m_b = i|n_b)$, the prior probability of a bag of size n_b having i positive instances. We estimate $P(m_b = i|n_b)$ using $P(m_b > 0|n_b)$ and making assumptions about the number of positive instances in positive bags. For instance, for the protein-interaction task we assume a uniform distribution for the number of positive instances in a positive bag: $P(m_b = i|n_b) = \frac{P(m_b>0|n_b)}{n_b-1}$ for $i > 0$. This reflects our belief that any number of positive instances is possible. For the protein-lexicon task we assume that in most cases if word w is a protein name then all its occurrences are positive[2]: $P(m_b = n_b|n_b) = P(m_b > 0|n_b)$ and $P(m_b = i|n_b) = 0$ for $i = 1, \ldots, n_b - 1$. The above assumptions are based on conclusions drawn from looking at the distribution of positive instances in positive bags in the training set.

A key characteristic of these natural language tasks is the mismatch in the bag sizes between the labeled and the test corpus. For instance, consider the case where we have a labeled corpus of a few thousand abstracts and all of MEDLINE as a test corpus. The number of co-occurrences of two words w_1 and w_2 will be a lot different in the two corpora. Also, $P(m_b > 0|n_b)$, the prior probability that a bag with size m_b is positive, will differ between the two corpora. To estimate $P(m_b > 0|n_b)$, the prior probability of a bag of size n_b being positive we use Bayes' rule: $P(m_b > 0|n_b) = \frac{P(n_b|m_b>0)P(m_b>0)}{P(n_b)}$. $P(n_b)$, the probability of bag b having size n_b, is computed by counting the size of all bags throughout the test corpus[3]. $P(n_b|m_b > 0)$, the probability of a positive bag having size n_b, can be estimated by sampling the test corpus using the positive bags of the labeled corpus (assuming there are common bags between the two corpora, a reasonable assumption for the tasks we consider). Finally, we need to estimate the prior probability of a bag being positive $P(m_b > 0)$. Let $B(D)$ and $B(T)$ denote the collection of bags and $I(D)$ and $I(T)$ denote the collections of instances of the labeled set D and test set T. $B^+(D)$, $B^+(T)$, $I^+(D)$ and $I^+(T)$ denote the collections of *positive* bags and instances. $P(m_b > 0) = |B^+(T)|/|B(T)|$. $|B(T)|$ can be counted. We compute $|I^+(T)| = |I(T)||I^+(D)|/|I(D)|$, assuming the same rate of positive instances in the labeled and test corpus. Finally, using $P(n_b|m_b > 0)$ we estimate the number of positive bags by solving $|I^+(T)| = \sum_{i=0}^{|B^+(T)|} P(n_b = i|m_b > 0)i$ for $|B^+(T)|$.

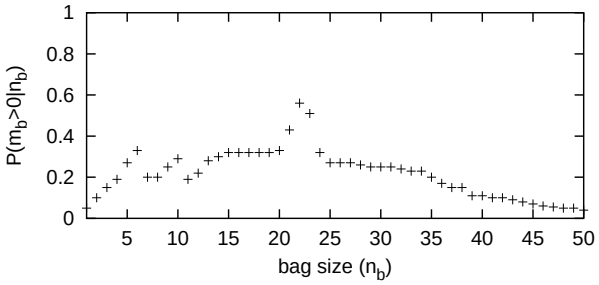
The use of priors $P(m_b > 0|n_b)$ estimated using this procedure is called the *estimated class priors* scheme. Because the estimation of these probabilities is complex (and in some rare cases even impossible), in our experimental evaluation we also consider using $P(m_b > 0|n_b) = P(m_b = 0|n_b) = 0.5$, a scheme which we call *uniform class priors*. Our empirical results suggest that BECIP has high accuracy even with the use of the inaccurate uniform priors. Thus, estimating the priors can be avoided if needed.

Figure 1 shows plots of the estimated prior probability $P(m_b > 0|n_b)$ of a bag being positive versus the bag size used in the experiments described in Section 4. The bag sizes for which $P(m_b > 0|n_b)$ becomes large are those in the middle of the x range. Small bag sizes represent bags with uncommon words that appear infrequently in the dataset and hence the prior probability is small since we expect to find a protein-interaction fact or a protein at least a few times in the dataset. Large bags are mostly bags with common words (such as "the" or "gene"), which are unlikely to be positive.

An underlying assumption of these estimation procedures is that statistics about the whole test corpus are available to us. For most tasks this is easy to satisfy. For instance, indexing all of MEDLINE is not trivial but is within the capabilities of current indexing technology [10]. When faced with



(a) Protein interaction



(b) Protein lexicon

Figure 1: Plots of the prior probability $P(m_b > 0|n_b)$ of a bag being positive versus the bag size n_b for the protein-interaction task and the protein-lexicon task. Note the difference in the x and y axes between the two plots. We only show the priors for small bag sizes, and note that for larger bags the prior probability of a bag being positive is negligible. For the protein-lexicon task only nouns and adjectives are candidate bags, hence the relatively high $P(m_b > 0|n_b)$ probability for some n_b.

[2]Note that according to the task definition, a bag is positive if and only if it contains at least one positive instance. The use of $P(m_b = n_b|n_b) = P(m_b > 0|n_b)$ as the priors for the number of positive instances in a bag simply reflects our expectations about the particular application and does not change the task definition.

[3]We note that this can be done without violating the standard machine learning assumptions about test sets (corpora). Indeed, the labels of test instances may be unknown but the test set can still be used in suitable ways. This practice is also common in the research field of transduction.

frequently changing corpora, such as news articles which are extended on a daily basis, incremental indexing is required to include new documents. Note that documents need not be stored but may be discarded once indexed. Finally, applying our techniques on a collection of web pages is also quite straightforward. In fact, to estimate the priors we need statistics similar to those kept by search engines.

4. EXPERIMENTAL EVALUATION

In this section we present our experiments in order to determine (i) the accuracy of `BECIP` predictions compared to the baseline methods, (ii) the accuracy of `BECIP` without using *estimated* class priors, and (iii) the behavior of the various evidence combination methods under different values for the bag-size mean and variance in the corpus.

4.1 Experimental Setup

We evaluate our approaches on the two natural language tasks described in Section 1: identifying protein interactions and building a lexicon of protein names. We use the same corpus for both tasks. The corpus is a collection of abstracts of yeast-related scientific papers collected from MEDLINE. It consists of 5,728 abstracts containing 47,473 sentences. The dataset is labeled with 1,503 unique protein-interaction assertions, with 8,088 total occurrences. Not all protein names are labeled in the dataset, so we do not have accurate estimates of their numbers. We manually reviewed the words predicted as proteins by our classifier and found 4,382 protein names with approximately 60,000 total occurrences. To recognize and extract protein interactions, we train hidden Markov models (HMMs) which are given shallow-parsed sentences as input [12; 17]. To extract protein names we train decision-tree models using C5.0 [11]. The decision trees classify each word in the corpus as positive (protein) or negative (non-protein) using lexical and syntactic features that represent the context in which it appears. The features used in our representation include (i) the type of phrase in which the candidate word occurs and the types of neighboring phrases, (ii) the stem and part-of-speech of neighboring words, and (iii) grammatical patterns learned from each training set using the Autoslog algorithm [13].

After training each model, we run it on a test set, collect the set of extractions made and assemble them into bags. We then consider combining evidence for these predictions using `BECIP` and several baseline methods. Each evidence-combination method assigns a confidence $C_+(b)$ for each bag b b. By varying a threshold on these confidence values we construct precision-recall curves to compare the accuracy of the various combination methods. *Precision* is defined as the number of correct extractions (i.e. true positive bags) divided by the number of extractions made. *Recall* is defined as the number of correct extractions divided by the total number of positives in the data set. We conduct a five-fold cross-validation experiment and pool the results from each test set to construct the precision-recall graphs.

Note that all of the evidence-combination methods work with the same set of predictions. Therefore, all of the methods have the same endpoint precision and recall. However, the evidence-combination methods may differ in the confidences they assign to various predictions, and thus some methods may have better precision than others for lower values of recall. The endpoint recall is normalized to one in our curves because the endpoint is the same for all evidence combination methods. Moreover, for the protein-lexicon task, we do not know the true recall level since we do not know the true number of proteins in the data set,

The bags with at least one positive prediction for the protein-interaction task have mean size of 6.8 and standard deviation of 10.17. For the protein-lexicon task the mean bag size is 50.25 and the standard deviation is 153.72. The estimated true positive and false positive rates of the classifier for the protein-interaction task are $t = 0.39$ and $f = 0.0002$. For the protein-lexicon task they are $t = 0.35$ and $f = 0.019$.

4.2 Baseline Methods for Evidence Combination

We empirically compare `BECIP` against several baseline evidence-combination methods. Some of these methods, such as `Soft-OR` and `Noisy-OR`, are standard methods for combining evidence. Others, e.g., `Weighted-Majority` and `Soft-Count`, are based on our intuitions about what schemes may be effective for evidence combination in the tasks we consider.

We assume that for each instance for which the classifier predicts the positive class it also assigns a confidence $C_+(e_i)$ between zero and one (to which we assign probability semantics) which captures the classifier's belief in that prediction. b^+ denotes the set of instances of b classified as positive.

Soft-OR (SO) Under this approach, bag b is assigned the positive class with confidence equal to the confidence of the most confident instance predicted as positive:

$$C_+(b) \;=\; \max_i\{C_+(e_i)\}, \forall\, e_i \;\in\; b^+.$$

Noisy-OR (NO) This approach assumes that all predictions are independent and assigns confidence to the bag class equal to the probability of *at least* one of the instances being positive:

$$C_+(b) \;=\; 1 - \prod_i [1 - C_+(e_i)],\;\; \forall e_i \in b^+.$$

Soft-Count (SC) This approach, which is a soft version of counting the number of positive predictions that also takes into account their confidences, assigns the positive class to bag b with confidence equal to:

$$C_+(b) \;=\; \sum_i C_+(e_i),\;\; \forall e_i \in b^+.$$

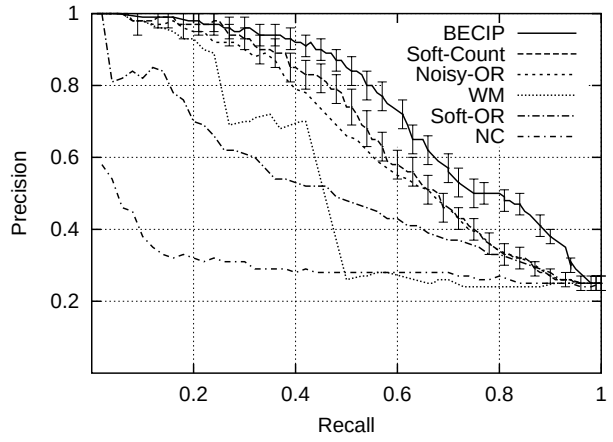
Note that with this scheme, $C_+(b)$ does not have probabilistic semantics.

Weighted-Majority (WM) The Weighted Majority approach assigns confidence to the positive class equal to:

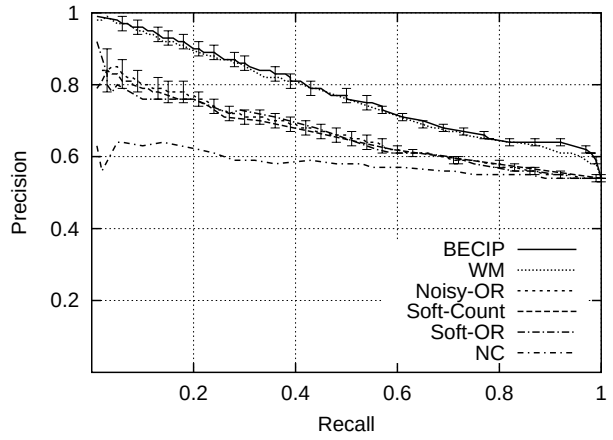
$$C_+(b) \;=\; |b^+|^2/|b|.$$

It is motivated by the simple Majority-Voting scheme ($C_+(b) = |b^+|/|b|$), with the difference that the numerator is multiplied by $|b^+|$ to increase the confidence for bags with many positive predictions. The simple Majority-Voting scheme performs poorly and is omitted from the experimental section.

No evidence combination (NC) In this approach, the confidence associated with the positive class for bag b is equal to the confidence of a randomly chosen positive



(a) Protein interactions



(b) Protein lexicon

Figure 2: Precision-recall graphs for the protein-interaction extraction and the protein-lexicon building tasks. 95% Confidence Intervals are shown for the two best methods. Recall has been normalized to one. For clarity, the key shows the method titles in order of highest precision.

prediction for that bag. This baseline is useful for determining the expected decrease in accuracy when evidence is not combined across predictions using any method.

The first three approaches, SO, NO and SC, ignore the size of the bag and focus on the positive predictions. All three can suffer from very large bags with many negative instances, as the false positive predictions on those can inflate the confidence assigned to that bag. They are expected to perform well on tasks with small bag size mean and variance. WM tries to balance the number of positive predictions and the bag size and is better suited to applications with large bag size mean and variance.

4.3 Accuracy of BECIP

Figure 2 shows the precision-recall graphs for the protein-interaction and the protein tasks. In the protein-interaction task, BECIP achieves significantly higher precision (at the 95% level) from Soft-Count, the second best method, for recall values between 0.38 and 0.94. For the protein-lexicon task, the precision of BECIP is identical to that of Weighted-Majority. They both are significantly more accurate than all other methods. For BECIP, we used the *estimated class priors* scheme.

	n_b		BECIP	Soft OR	Noisy OR	Soft Count	Weight Maj.
Max	$Mean$	$StDev$					
5	3.42	1.67	0.66	0.69	0.69	0.69	0.7
10	5.34	3.61	0.71	0.67	0.70	0.70	0.69
20	7.96	7.12	0.71	0.66	0.68	0.68	0.69
30	9.75	10.17	0.71	0.65	0.67	0.67	0.69
50	12.17	15.22	0.72	0.65	0.65	0.65	0.70
100	15.86	25.09	0.71	0.62	0.63	0.63	0.69
200	19.86	39.62	0.72	0.61	0.63	0.63	0.69
300	22.41	51.46	0.72	0.61	0.62	0.62	0.69
500	26.02	71.20	0.71	0.61	0.62	0.62	0.69
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)

Table 3: The accuracy of the various evidence combination methods in terms of the precision-recall break-even point for different bag size averages and variances. Each line represents different bag size settings. Column (a) shows the maximum bag size. The first three columns show the bag size mean and standard deviation. The remaining columns show the precision-recall break-even point of the various evidence combination methods.

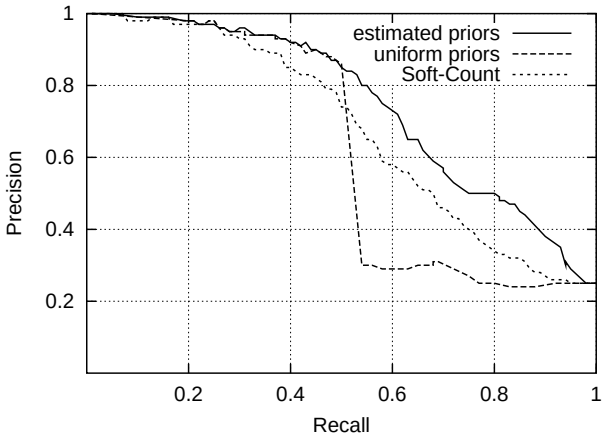
All methods for evidence combination have significantly better precision-recall curves than *no evidence combination (NC)*. Among the evidence combination methods, BECIP has the highest precision for all values of recall in both tasks. WM matches the precision of BECIP on the protein-lexicon task but performs poorly on the protein-interaction task. On the contrary, SC which is the best baseline for the protein-interaction task has relatively poor precision on the protein task. WM performs well in the protein-lexicon task because its bias matches the fact that typically, most instances in a bag have the same label as the bag. This is not true in the protein-interaction task, in which WM is outperformed by NO and SC whose biases are more appropriate for this task. SO has overall mediocre precision as it is not aggressive enough in combining evidence.

4.4 BECIP with Uniform Class Priors

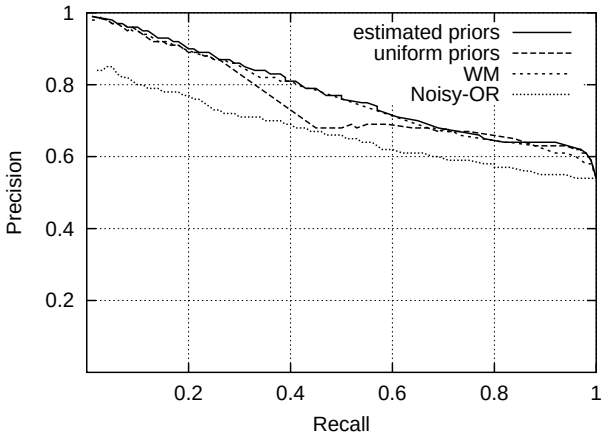
We also apply BECIP using uniform priors (see *uniform class priors* scheme in Section 3). Figure 3 shows the accuracy of BECIP with estimated and uniform priors. The graphs also show the best baseline for each task, for reasons of comparison. The lower precision of the uniform-prior variant is due mostly to small bags with few positive predictions, for which the estimated prior $P(m_b > 0|n_b)$ is small, something that is not taken into account by the uniform priors. For large bags, the positive and negative predictions overwhelm the contribution of the priors in calculating $P(m_b > 0|k_b, n_b)$. Indeed, for both tasks if bags with size less than or equal to three ($n_b \leq 3$) are ignored the precision of the two schemes is practically identical, as shown in Figure 4.

4.5 Varying the Bag Size Mean and Variance

We also evaluate the behavior of the various evidence combination methods under conditions of different average bag size and bag size variance. We use the bags from the protein extraction task to simulate different conditions. We did not perform this experiment on the protein-interaction task as the bag sizes have small mean and variance.

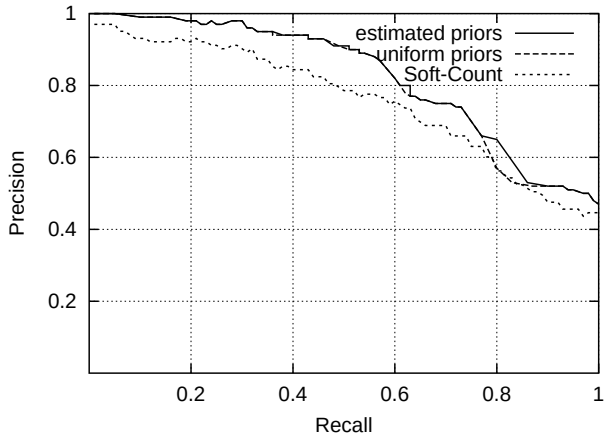


(a) Protein interactions

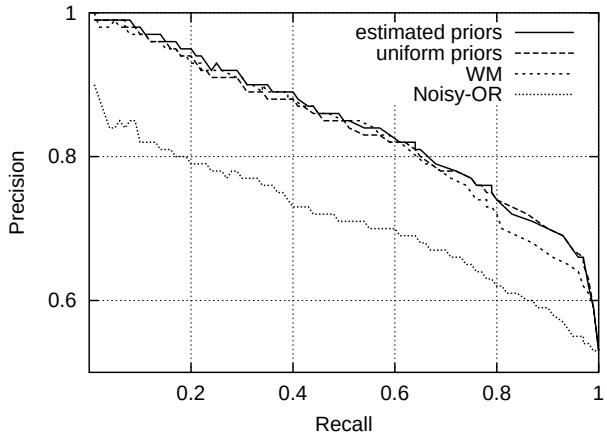


(b) Protein lexicon

Figure 3: BECIP's accuracy with *estimated* and *uniform* priors for the two tasks. For comparison, the graphs also show the best baselines for each task.



(a) Protein interactions



(b) Protein lexicon

Figure 4: BECIP's accuracy with *estimated* and *uniform* priors, for bags with size larger than three ($n_b > 3$) for the two tasks. For comparison, the graphs also show the best baselines for each task. Note that the y-axis for the protein-lexicon task ranges from .5 to 1.

For each experiment, we set an upper bound m_i on the bag size (shown in column (a) of Table 3) and discard any extra instances from bags with size greater than m_i. This "trimming" results in different values for the mean and standard deviation of the bag sizes (shown in columns (b) and (c) of Table 3). We evaluate each method using the Precision-Recall break-even point (i.e., the point where precision and recall are equal). Columns (d) to (h) of Table 3 show the accuracy of each evidence combination method. BECIP used the *uniform class priors* scheme.

Each line in Table 3 represents a different experiment. Recall is not the same for each experiment, because for some of the bags it happens that all of the positive predictions are discarded to satisfy the maximum size constraint. The larger the bag size, the larger the recall for all evidence methods. We have again normalized recall to one, since all methods have the same recall for a given experiment. As a result, only comparisons within rows are valid, but not comparisons within columns.

The observed results validate our hypotheses about the expected behavior of each method. BECIP outperforms all other methods for most settings and has consistently the best precision at each recall point. WM outperforms all other baselines as the bag size mean and standard deviation increase. SO, NO and SC have good accuracy for relatively small bag mean (3-10 average bag size) and variance, while their accuracy degrades for larger and more variable bag sizes. BECIP did not perform well for maximum bag size 5, because the uniform priors are very inaccurate for this setting.

5. CONCLUSIONS

In this paper we have addressed the issue of evidence combination for natural language tasks. In such tasks, information often appears in multiple contexts with different meaning. Current practice employs methods that make decisions based on document passages of limited length, resulting in a need to combine these multiple decisions. We have presented a formal definition of this problem and developed BECIP, a theoretically sound method for combining evidence based on abstracting the behavior of these natural language methods as Bernoulli trials. We have also considered a number of simpler baseline methods. Our experimental evaluation indicates that combining evidence can improve precision in

these tasks. Furthermore, we have shown that BECIP outperforms the baselines in the most cases, and we have identified situations where the use of the simpler baselines may be more suitable.

6. ACKNOWLEDGMENTS

This research was supported in part by NIH grant 1R01 LM07050-01, NSF grant IIS-0093016, and a grant to the University of Wisconsin Medical School under the Howard Hughes Medical Institute Research Resources Program for Medical Schools.

7. REFERENCES

[1] C. Blaschke, M. Andrade, C. Ouzounis, and A. Valencia. Automatic extraction of biological information from scientific text: Protein-protein interactions. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 60–67, Heidelberg, Germany, 1999. AAAI Press.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100. ACM Press, 1998.

[3] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *Working Papers of the ACL-97 Workshop on Natural Language Learning*, 1997.

[4] T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[5] L. Hirschman, J. Park, J. Tsujii, L. Wong, and C. Wu. Accomplishments and challenges in literature data mining for biology. *Bioinformatics*, 18:1553–1561, 2002.

[6] M. Krauthammer, P. Kra, I. Iossifov, S. Gomez, G. Hripcsak, V. Hatzivassiloglou, C. Friedman, and A. Rzhetsky. Of truth and pathways: Chasing bits of information through myriads of articles. *Bioinformatics*, 18(Suppl. 1):S249–S257, 2002.

[7] B. Magnini, M. Negri, R. Prevete, and H. Tanev. Is it the right answer? Exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 425–432, Philadelphia, 2002.

[8] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598, Stanford, CA, 2000. Morgan Kaufmann.

[9] I. Muslea. *Active Learning with Multiple Views*. PhD thesis, Department of Computer Science, University of Southern California, Los Angeles, CA, 2002.

[10] National Library of Medicine. The MEDLINE database, 2003. http://www.ncbi.nlm.nih.gov/PubMed/.

[11] J. R. Quinlan. C5.0 decision tree software, 1999. http://www.rulequest.com/.

[12] S. Ray and M. Craven. Representing sentence structure in hidden Markov models for information extraction. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1273–1279, Seattle, WA, 2001. Morgan Kaufmann.

[13] E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049, Portland, OR, 1996. AAAI/MIT Press.

[14] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, Orlando, FL, 1999. AAAI Press.

[15] D. Roth and W. Yih. Probabilistic reasoning for entity and relation recognition. In *Proceedings of the Nineteenth International Conference on Computational Linguistics*, Taipei, Taiwan, 2002. Morgan Kaufmann.

[16] H. Shatkay, S. Edwards, W. J. Wilbur, and M. Boguski. Genes, themes and microarrays: Using information retrieval for large-scale gene analysis. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 317–328, La Jolla, CA, 2000. AAAI Press.

[17] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden Markov models for information extraction. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003. Morgan Kaufmann.

[18] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 1999.

[19] A. Yeh, L. Hirschman, and A. Morgan. Background and overview for KDD Cup 2002 task 1: Information extraction from biomedical articles. *SIGKDD Explorations*, 4(2):87–89, 2003.