Minority Vote: At-Least-N Voting Improves Recall for Extracting Relations

Nanda Kambhatla
IBM T.J. Watson Research Center
1101 Kitchawan Road Rt 134
Yorktown, NY 10598
nanda@us.ibm.com

Abstract

Several NLP tasks are characterized by asymmetric data where one class label *NONE*, signifying the absence of any structure (named entity, coreference, relation, etc.) dominates all other classes. Classifiers built on such data typically have a higher precision and a lower recall and tend to overproduce the *NONE* class. We present a novel scheme for voting among a committee of classifiers that can significantly boost the recall in such situations. We demonstrate results showing up to a 16% relative improvement in ACE value for the 2004 ACE relation extraction task for English, Arabic and Chinese.

1 Introduction

Statistical classifiers are widely used for diverse NLP applications such as part of speech tagging (Ratnaparkhi, 1999), chunking (Zhang et al., 2002), semantic parsing (Magerman, 1993), named entity extraction (Borthwick, 1999; Bikel et al., 1997; Florian et al., 2004), coreference resolution (Soon et al., 2001), relation extraction (Kambhatla, 2004), etc. A number of these applications are characterized by a dominance of a *NONE* class in the training examples. For example, for coreference resolution, classifiers might classify whether a given pair of mentions are references to the same entity or not. In this case, we typically have a lot more examples of mention pairs that are not coreferential (i.e. the *NONE* class) than otherwise. Similarly, if a classifier is predicting the presence/absence of a semantic relation between two mentions, there are typically far more examples signifying an absence of a relation.

Classifiers built with asymmetric data dominated by one class (a *NONE* class donating absence of a relation or coreference or a named entity etc.) can overgenerate the *NONE* class. This often results in a unbalanced classifier where precision is higher than recall.

In this paper, we present a novel approach for improving the recall of such classifiers by using a new voting scheme from a committee of classifiers. There are a plethora of algorithms for combining classifiers (e.g. see (Xu et al., 1992)). A widely used approach is a **majority voting** scheme, where each classifier in the committee gets a vote and the class with the largest number of votes 'wins' (i.e. the corresponding class is output as the prediction of the committee).

We are interested in improving overall recall and reduce the overproduction of the class *NONE*. Our scheme predicts the class label C obtaining the second highest number of votes when *NONE* gets the highest number of votes, provided C gets **at least** N votes. Thus, we predict a label other than *NONE* when there is some evidence of the presense of the structure we are looking for (relations, coreference, named entities, etc.) even in the absense of a clear majority.

This paper is organized as follows. In section 2, we give an overview of the various schemes for combining classifiers. In section 3, we present our vot-

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 460–466,
Sydney, July 2006. ©2006 Association for Computational Linguistics

ing algorithm. In section 4, we describe the ACE relation extraction task. In section 5, we present empirical results for relation extraction and we discuss our results and conclude in section 6.

2 Combining Classifiers

Numerous methods for combining classifiers have been proposed and utlized to improve the performance of different NLP tasks such as part of speech tagging (Brill and Wu, 1998), identifying base noun phrases (Tjong Kim Sang et al., 2000), named entity extraction (Florian et al., 2003), etc. Ho *et al* (1994) investigated different approaches for reranking the outputs of a committee of classifiers and also explored union and intersection methods for reducing the set of predicted categories. Florian *et al* (2002) give a broad overview of methods for combining classifiers and present empirical results for word sense disambiguation.

Xu *et al* (1992) and Florian *et al* (2002) consider three approaches for combining classifiers. In the first approach, individual classifiers output posterior probabilities that are merged (e.g. by taking an average) to arrive at a composite posterior probability of each class. In the second scheme, each classifier outputs a ranked list of classes instead of a probability distribution and the different ranked lists are merged to arrive at a final ranking. Methods using the third approach, often called *voting methods*, treat each classifier as a black box that outputs only the top ranked class and combines these to arrive at the final decision (class). The choice of approach and the specific method of combination may be constrained by the specific classification algorithms in use.

In this paper, we focus on voting methods, since for small data sets, it is hard to reliably estimate probability distributions or even a complete ordering of classes especially when the number of classes is large.

A widely used voting method for combining classifiers is a **Majority Vote** scheme (e.g. (Brill and Wu, 1998; Tjong Kim Sang et al., 2000)). Each classifier gets to vote for its top ranked class and the class with the highest number of votes ’wins’. Henderson *et al* (1999) use a Majority Vote scheme where different parsers vote on constituents’ membership in a hypothesized parse. Halteren *et al* (1998) compare a number of voting methods including a Majority Vote scheme with other combination methods for part of speech tagging.

In this paper, we induce multiple classifiers by using **bagging** (Breiman, 1996). Following Breiman’s approach, we obtain multiple classifiers by first making bootstrap replicates of the training data and training different classifiers on each of the replicates. The bootstrap replicates are induced by repeatedly *sampling with replacement* training events from the original training data to arrive at replicate data sets of the same size as the training data set. Breiman (1996) uses a Majority Vote scheme for combining the output of the classifiers. In the next section, we will describe the different voting schemes we explored in our work.

3 At-Least-N Voting

We are specifically interested in NLP tasks characterized by asymmetric data where, typically, we have far more occurances of a *NONE* class that siginifies the absense of structure (e.g. a named entity, or a coreference relation or a semantic relation). Classifiers trained on such data sets can overgenerate the *NONE* class, and thus have a higher precision and lower recall in discovering the underlying structure (i.e. the named entities or coreference links etc.). With such tasks, the benefits yielded by a Majority Vote is limited, since, because of the asymmetry in the data, a majority of the classifiers might predict *NONE* most of the time.

We propose alternative voting schemes, dubbed **At-Least-N Voting**, to deal with the overproduction of *NONE*. Given a committee of classifiers (obtained by bagging or some other mechanism), the classifiers first cast their vote. If the majority vote is for a class C other than *NONE,* we simply output C as the prediction. If the majority vote is for *NONE*, we output the class label obtaining the second highest number of votes, *provided* it has at least N votes. Thus, we choose to defer to the minority vote of classifiers which agree on finding some structure even when the majority of classifiers vote for *NONE*. We expect this voting scheme to increase recall at the expense of precision.

At-Least-N Voting induces a spectrum of combination methods ranging from a Majority Vote (when N is more than half of the total number of classifiers) to a scheme, where the evidence of any structure by even one classifier is believed (At-Least-1 Voting). The exact choice of N is an empirical one and depends on the amount of asymmetry in the data and the imbalance between precision and recall in the classifiers.

4 The ACE Relation Extraction Task

Automatic Content Extraction (ACE) is an annual evaluation conducted by NIST (NIST, 2004) on information extraction, focusing on extraction of entities, events, and relations. The Entity Detection and Recognition task entails detection of mentions of entities and grouping together the mentions that are references to the same entity. In ACE terminology, *mentions* are references in text (or audio, chats, ...) to real world *entities*. Similarly *relation mentions* are references in text to semantic relations between entity mentions and *relations* group together all relation mentions that identify the same semantic relation between the same entities.

In the frament of text:

> John’s son, Jim went for a walk. Jim liked his father.

all the underlined words are mentions referring to two entities, John, and Jim. Morover, John and Jim have a *family* relation evidenced as two relation mentions ”John’s son” between the entity mentions ”John” and ”son” and ”his father” between the entity mentions ”his” and ”father”.

In the relation extraction task, systems must predict the presence of a predetermined set of binary relations among mentions of entities, label the relation, and identify the two arguments. In the 2004 ACE evaluation, systems were evaluated on their efficacy in correctly identifying relations among both system output entities and with ’true’ entities (i.e. as annotated by human annotators as opposed to system output). In this paper, we present results for extracting relations between ’true’ entities.

Table 1 shows the set of relation types, subtypes, and their frequency counts in the training data for the 2004 ACE evaluation. For training classifiers, the great paucity of positive training events (where relations exist) compared to the negative events (where relations do not exist) suggest that schemes for improving recall might benefit this task.

Type	**Subtype**	**Count**
ART	*user-or-owner*	*140*
(agent artifact)	*inventor/manufacturer*	*3*
	other	*6*
EMP-ORG	*employ-executive*	*420*
	employ-staff	*416*
	employ-undetermined	*62*
	member-of-group	*126*
	partner	*11*
	subsidiary	*213*
	other	*37*
GPE-AFF	*citizen-or-resident*	*173*
(GPE affiliation)	*based-in*	*225*
	other	*63*
DISCOURSE	*-none-*	*122*
PHYSICAL	*located*	*516*
	near	*81*
	part-whole	*333*
PER-SOC	*business*	*119*
(personal/social)	*family*	*115*
	other	*28*
OTHER-AFF	*ethnic*	*28*
(PER/ORG	*ideology*	*26*
affiliation)	*other*	*27*

Table 1: The set of types and subtypes of relations used in the 2004 ACE evaluation.

5 Experimental Results

In this section, we present results of experiments comparing three different methods of combining classifiers for ACE relation extraction:

- *At-Least-N* for different values of N,
- Majority Voting, and
- a simple algorithm, called **summing**, where we add the posterior scores for each class from all the classifiers and select the class with the maximum summed score.

Since the official ACE evaluation set is not publicly available, to facilitate comparison with our results and for internal testing of our algorithms, for each language (English, Arabic, and Chinese), we

	En	**Ar**	**Ch**
Training Set (documents)	*227*	*511*	*480*
Training Set (rel-mentions)	*3290*	*4126*	*4347*
Test Set (documents)	*114*	*178*	*166*
Test Set (rel-mentions)	*1381*	*1894*	*1774*

Table 2: The Division of LDC annotated data into training and development test sets.

divided the ACE 2004 training data provided by LDC in a roughly 75%:25% ratio into a training set and a test set. Table 2 summarizes the number of documents and the number of relation mentions in each data set. The test sets were deliberately chosen to be the most recent 25% of documents in chronological order, since entities and relations in news tend to repeat and random shuffles can greatly reduce the out-of-vocabulary problem.

5.1 Maximum Entropy Classifiers

We used bagging (Breiman, 1996) to create replicate training sets of the same size as the original training set by repeatedly sampling with replacement from the training set. We created 25 replicate training sets (bags) for each language (Arabic, Chinese, English) and trained separate maximum entropy classifiers on each bag. We then applied At-Least-N (N = 1,2,5), Majority Vote, and Summing algorithms with the trained classifiers and measured the resulting performance on our development set.

For each bag, we built maximum entropy models to predict the presence of relation mentions and the type and subtype of relations, when their presence is predicted. Our models operate on every pair of mentions in a document that are not references to the same entity, to extract relation mentions. Since there are 23 unique type-subtype pairs in Table 1, our classifiers have 47 classes: two classes for each pair corresponding to the two argument orderings (e.g. ”John’s son” vs. ”his father”) and a *NONE* class signifying no relation.

Similar to our earlier work (Kambhatla, 2004), we used a combination of lexical, syntactic, and semantic features including all the words in between the two mentions, the entity types and subtypes of the two mentions, the number of words in between the two mentions, features derived from the smallest parse fragment connecting the two mentions, etc. These features were held constant throughout these experiments.

5.2 Results

We report the F-measure, precision and recall for extracting relation mentions for all three languages. We also report *ACE value*[1], the official metric used by NIST that assigns 0% value to a system that produces no output and a 100% value to a system that extracts all relations without generating any false alarms. Note that the ACE value counts each relation only once even if it is expressed in text many times as different relation mentions. The reader is referred to the NIST web site (NIST, 2004) for more details on the ACE value computation.

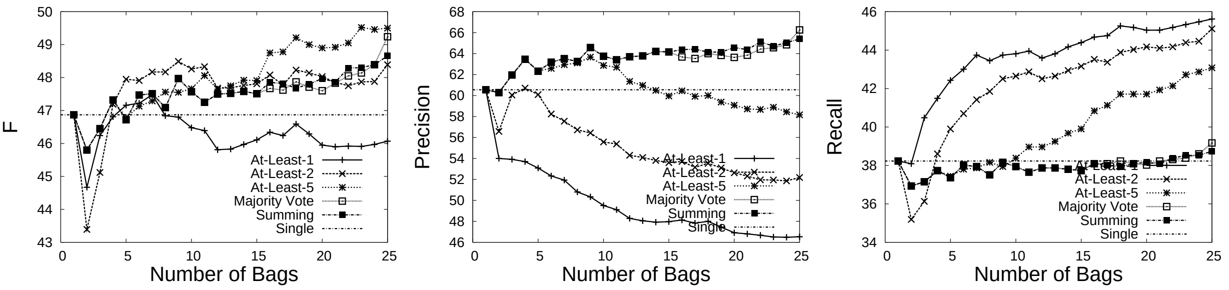
Figures 1(a), 1(b), and 1(c) show the F-measure, precision, and recall respectively for the English test set obtained by different classifier combination techniques as we vary the number of bags. Figures 2(a), 2(b), and 2(c) show similar curves for Chinese, and Figures 3(a), 3(b), and 3(c) show similar curves for Arabic. All these figures show the performance of a single classifier as a straight line.

From the plots, it is clear that our hope of increasing recall by combining classifiers is realized for all three languages. As expected, the recall rises fastest for At-Least-N when N is small, i.e when small minority opinion or even a single dissenting opinion is being trusted. Of course, the rise in recall is at the expense of a loss of precision. Overall, At-Least-N for intermediate ranges of N (N=5 for English and Chinese and N=2 for Arabic) performs best where the moderate loss in precision is more than offset by a rise in recall.

Both the Majority Vote method and the Summing method succeed in avoiding a sharp loss of precision. However, they fail to increase the recall significantly either.

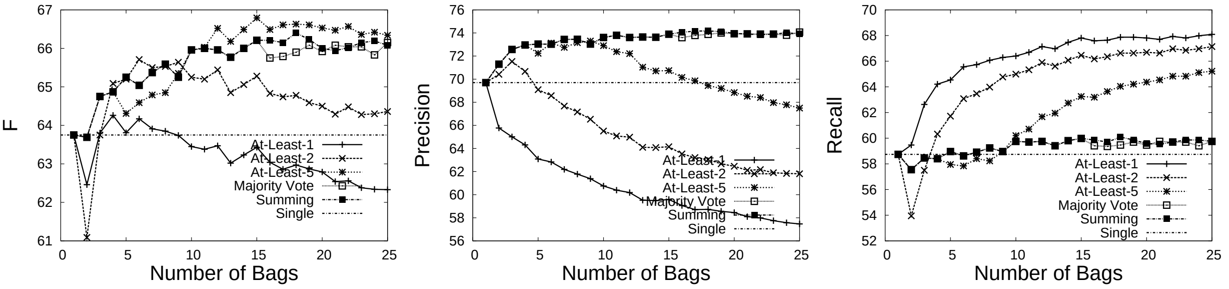
Table 3 summarizes the best results (F-measure) for each classifier combination method for all three languages compared with the result for a single classifier. At their best operating points, all three combination methods handily outperform the single classifier. At-Least-N seems to have a slight edge over the other two methods, but the difference is small.

[1] Here we use the ACE value metric used for the ACE 2004 evaluation



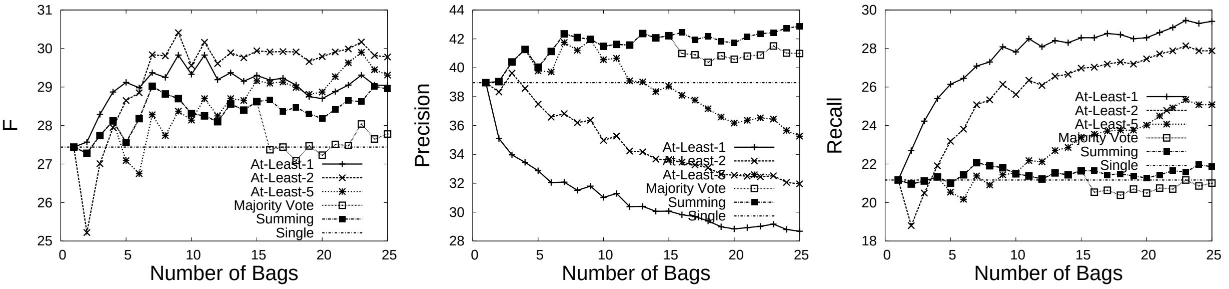
(a) F-measure (b) Precision (c) Recall

Figure 1: Comparing F-measure, precision, and recall of different voting schemes for **English** relation extraction.



(a) F-measure (b) Precision (c) Recall

Figure 2: Comparing F-measure, precision, and recall of different voting schemes for **Chinese** relation extraction.



(a) F-measure (b) Precision (c) Recall

Figure 3: Comparing F-measure, precision, and recall of different voting schemes for **Arabic** relation extraction.

	English	**Arabic**	**Chinese**
Single	*46.87*	*27.47*	*63.75*
At-Least-N	**49.52**	**30.41**	**66.79**
Majority Vote	*49.24*	*29.02*	*66.21*
Summing	*48.66*	*29.02*	*66.40*

Table 3: Comparing the best F-measure obtained by At-Least-N Voting with Majority Voting, Summing and the single best classifier.

	English	**Arabic**	**Chinese**
Single	*59.6*	*37.3*	*69.6*
At-Least-N	**63.9**	**43.5**	**71.0**

Table 4: Comparing the ACE Value obtained by At-Least-N Voting with the single best classifier for the operating points used in Table 3.

Table 4 shows the ACE value obtained by our best performing classifier combination method (At-Least-N at the operating points in Table 3) compared with a single classifier. Note that while the improvement for Chinese is slight, for Arabic performance improves by over 16% relative and for English, the improvement is over 7% relative over the single classifier[2]. Since the ACE value collapses relation mentions referring to the same relation, finding new relations (i.e. recall) is more important. This might explain the relatively larger difference in ACE value between the single classifier performance and At-Least-N.

The rules of the ACE evaluation prohibit us from presenting a detailed comparison of our relation extraction system with the other participants. However, our relation extraction system (using the At-Least-N classifier combination scheme as described here) performed very competitively in 2004 ACE evaluation both in the system output relation extraction task (RDR) and the relation extraction task where the 'true' mentions and entities are given.

Due to time limitations, we did not try At-Least-N with $N > 5$. From the plots, there is a potential for getting greater gains by experimenting with a larger number of bags and with a larger N.

6 Discussion

Several NLP problems exhibit a dominance of a *NONE* class that typically signifies a lack of structure like a named entity, coreference, etc. Especially when coupled with small training sets, this results in classifiers with unbalanced precision and recall. We have argued that a classifier voting scheme that is focused on improving recall can help increase overall performance in such situations.

We have presented a class of voting methods, dubbed *At-Least-N* that defer to the opinion of a minority of classifiers (consisting of N members) even when the majority predicts *NONE*. This can boost recall at the expense of precision. However, by varying N and the number of classifiers, we can pick an operating point that improves the overall F-measure.

We have presented results for ACE relation extraction for three languages comparing At-Least-N with Majority Vote and Summing methods for combining classifiers. All three classifier combination methods significantly outperform a single classifier. Also, At-Least-N consistently gave us the best performance across different languages.

We used bagging to induce multiple classifiers for our task. Because of the random bootstrap sampling, different replicate training sets might tilt towards one class or another. Thus, if we have many classifiers trained on the replicate training sets, some of them are likely to be better at predicting certain classes than others. In future, we plan to experiment with other methods for collecting a committee of classifiers.

References

D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of ANLP-97*, pages 194–201.

A. Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, New York University.

L. Breiman. 1996. Bagging predictors. In *Machine Learning*, volume 24, page 123.

E. Brill and J. Wu. 1998. Classifier combination for improved lexical disambiguation. *Proceedings of COLING-ACL'98*, pages 191–195, August.

[2] Note that ACE value metric used in the ACE 2004 evaluation weights entitites differently based on their type. Thus, relations with PERSON-NAME arguments end up contributing a lot more the overall score than relations with FACILITY-PRONOUN arguments.

Radu Florian and David Yarowsky. 2002. Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of EMNLP'02*, pages 25–32.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of CoNNL'03*, pages 168–171.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N Nicolov, and S Roukos. 2004. A statistical model for multilingual entity detection and tracking. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 1–8.

J. Henderson and E. Brill. 1999. Exploiting diversity in natural language processing: Combining parsers. In *Proceedings on EMNLP99*, pages 187–194.

T. K. Ho, J. J. Hull, and S. N. Srihari. 1994. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, January.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *The Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 178–181, Barcelona, Spain, July. Association for Computational Linguistics.

D. Magerman. 1993. Parsing as statistical pattern recognition.

NIST. 2004. The ACE evaluation plan. www.nist.gov/speech/tests/ace/index.htm.

Adwait Ratnaparkhi. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–178.

W. M. Soon, H. T. Ng, and C. Y. Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

E. F. Tjong Kim Sang, W. Daelemans, H. Dejean, R. Koeling, Y. Krymolowsky, V. Punyakanok, and D. Roth. 2000. Applying system combination to base noun phrase identification. In *Proceedings of COLING 2000*, pages 857–863.

H. van Halteren, J. Zavrel, and W. Daelemans. 1998. Improving data driven wordclass tagging by system combination. In *Proceedings of COLING-ACL'98*, pages 491–497.

L. Xu, A. Krzyzak, and C. Suen. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. on Systems, Man. Cybernet*, 22(3):418–435.

T. Zhang, F. Damerau, and D. E. Johnson. 2002. Text chunking based on a generalization of Winnow. *Journal of Machine Learning Research*, 2:615–637.