Improving Name Tagging by Reference Resolution and Relation Detection

Heng Ji **Ralph Grishman**
Department of Computer Science
New York University
New York, NY, 10003, USA
hengji@cs.nyu.edu grishman@cs.nyu.edu

Abstract

Information extraction systems incorporate multiple stages of linguistic analysis. Although errors are typically compounded from stage to stage, it is possible to reduce the errors in one stage by harnessing the results of the other stages. We demonstrate this by using the results of coreference analysis and relation extraction to reduce the errors produced by a Chinese name tagger. We use an N-best approach to generate multiple hypotheses and have them re-ranked by subsequent stages of processing. We obtained thereby a reduction of 24% in spurious and incorrect name tags, and a reduction of 14% in missed tags.

1 Introduction

Systems which extract relations or events from a document typically perform a number of types of linguistic analysis in preparation for information extraction. These include name identification and classification, parsing (or partial parsing), semantic classification of noun phrases, and coreference analysis. These tasks are reflected in the evaluation tasks introduced for MUC-6 (named entity, coreference, template element) and MUC-7 (template relation).

In most extraction systems, these stages of analysis are arranged *sequentially*, with each stage using the results of prior stages and generating a single analysis that gets enriched by each stage. This provides a simple modular organization for the extraction system.

Unfortunately, each stage also introduces a certain level of error into the analysis. Furthermore, these errors are compounded – for example, errors in name recognition may lead to errors in parsing. The net result is that the final output (relations or events) may be quite inaccurate.

This paper considers how interactions between the stages can be exploited to reduce the error rate. For example, the results of coreference analysis or relation identification may be helpful in name classification, and the results of relation or event extraction may be helpful in coreference.

Such interactions are not easily exploited in a simple sequential model … if name classification is performed at the beginning of the pipeline, it cannot make use of the results of subsequent stages. It may even be difficult to use this information *implicitly*, by using features which are also used in later stages, because the representation used in the initial stages is too limited.

To address these limitations, some recent systems have used more *parallel* designs, in which a single classifier (incorporating a wide range of features) encompasses what were previously several separate stages (Kambhatla, 2004; Zelenko et al., 2004). This can reduce the compounding of errors of the sequential design. However, it leads to a very large feature space and makes it difficult to select linguistically appropriate features for particular analysis tasks. Furthermore, because these decisions are being made in parallel, it becomes much harder to express interactions between the levels of analysis based on linguistic intuitions.

Proceedings of the 43rd Annual Meeting of the ACL, pages 411–418,
Ann Arbor, June 2005.

In order to capture these interactions more explicitly, we have employed a sequential design in which multiple hypotheses are forwarded from each stage to the next, with hypotheses being reranked and pruned using the information from later stages. We shall apply this design to show how named entity classification can be improved by 'feedback' from coreference analysis and relation extraction. We shall show that this approach can capture these interactions in a natural and efficient manner, yielding a substantial improvement in name identification and classification.

2 Prior Work

A wide variety of trainable models have been applied to the name tagging task, including HMMs (Bikel et al., 1997), maximum entropy models (Borthwick, 1999), support vector machines (SVMs), and conditional random fields. People have spent considerable effort in engineering appropriate features to improve performance; most of these involve internal name structure or the immediate local context of the name.

Some other named entity systems have explored global information for name tagging. (Borthwick, 1999) made a second tagging pass which uses information on token sequences tagged in the first pass; (Chieu and Ng, 2002) used as features information about features assigned to other instances of the same token.

Recently, in (Ji and Grishman, 2004) we proposed a name tagging method which applied an SVM based on coreference information to filter the names with low confidence, and used coreference rules to correct and recover some names. One limitation of this method is that in the process of discarding many incorrect names, it also discarded some correct names. We attempted to recover some of these names by heuristic rules which are quite language specific. In addition, this single-hypothesis method placed an upper bound on recall.

Traditional statistical name tagging methods have generated a single name hypothesis. BBN proposed the N-Best algorithm for speech recognition in (Chow and Schwartz, 1989). Since then N-Best methods have been widely used by other researchers (Collins, 2002; Zhai et al., 2004).

In this paper, we tried to combine the advantages of the prior work, and incorporate broader knowledge into a more general re-ranking model.

3 Task and Terminology

Our experiments were conducted in the context of the ACE Information Extraction evaluations, and we will use the terminology of these evaluations:

entity: an object or a set of objects in one of the semantic categories of interest

mention: a reference to an entity (typically, a noun phrase)

name mention: a reference by name to an entity

nominal mention: a reference by a common noun or noun phrase to an entity

relation: one of a specified set of relationships between a pair of entities

The 2004 ACE evaluation had 7 types of entities, of which the most common were PER (persons), ORG (organizations), and GPE ('geo-political entities' – locations which are also political units, such as countries, counties, and cities). There were 7 types of relations, with 23 subtypes. Examples of these relations are "the CEO of Microsoft" (an *employ-exec* relation), "Fred's wife" (a *family* relation), and "a military base in Germany" (a *located* relation).

In this paper we look at the problem of identifying name mentions in Chinese text and classifying them as persons, organizations, or GPEs. Because Chinese has neither capitalization nor overt word boundaries, it poses particular problems for name identification.

4 Baseline System

4.1 Baseline Name Tagger

Our baseline name tagger consists of a HMM tagger augmented with a set of post-processing rules. The HMM tagger generally follows the Nymble model (Bikel et al, 1997), but with multiple hypotheses as output and a larger number of states (12) to handle name prefixes and suffixes, and transliterated foreign names separately. It operates on the output of a word segmenter from Tsinghua University.

Within each of the name class states, a statistical bigram model is employed, with the usual one-word-per-state emission. The various probabilities involve word co-occurrence, word features, and class probabilities. Then it uses A* search decoding to generate multiple hypotheses. Since these probabilities are estimated based on observations

seen in a corpus, "back-off models" are used to reflect the strength of support for a given statistic, as for the Nymble system.

We also add post-processing rules to correct some omissions and systematic errors using name lists (for example, a list of all Chinese last names; lists of organization and location suffixes) and particular contextual patterns (for example, verbs occurring with people's names). They also deal with abbreviations and nested organization names.

The HMM tagger also computes the margin – the difference between the log probabilities of the top two hypotheses. This is used as a rough measure of confidence in the top hypothesis (see sections 5.3 and 6.2, below).

The name tagger used for these experiments identifies the three main ACE entity types: Person (PER), Organization (ORG), and GPE (names of the other ACE types are identified by a separate component of our system, not involved in the experiments reported here).

4.2 Nominal Mention Tagger

Our nominal mention tagger (noun group recognizer) is a maximum entropy tagger trained on the Chinese TreeBank from the University of Pennsylvania, supplemented by list matching.

4.3 Reference Resolver

Our baseline reference resolver goes through two successive stages: first, coreference rules will identify some high-confidence positive and negative mention pairs, in training data and test data; then the remaining samples will be used as input of a maximum entropy tagger. The features used in this tagger involve distance, string matching, lexical information, position, semantics, etc. We separate the task into different classifiers for different mention types (name / noun / pronoun). Then we incorporate the results from the relation tagger to adjust the probabilities from the classifiers. Finally we apply a clustering algorithm to combine them into entities (sets of coreferring mentions).

4.4 Relation Tagger

The relation tagger uses a k-nearest-neighbor algorithm. For both training and test, we consider all pairs of entity mentions where there is at most one other mention between the heads of the two mentions of interest[1]. Each training / test example consists of the pair of mentions and the sequence of intervening words. Associated with each training example is either one of the ACE relation types or no relation at all. We defined a distance metric between two examples based on

- whether the heads of the mentions match
- whether the ACE types of the heads of the mentions match (for example, both are people or both are organizations)
- whether the intervening words match

To tag a test example, we find the k nearest training examples (where k = 3) and use the distance to weight each neighbor, then select the most common class in the weighted neighbor set.

To provide a crude measure of the confidence of our relation tagger, we define two thresholds, D_{near} and D_{far}. If the average distance d to the nearest neighbors $d < D_{near}$, we consider this a *definite* relation. If $D_{near} < d < D_{far}$, we consider this a *possible* relation. If $d > D_{far}$, the tagger assumes that no relation exists (regardless of the class of the nearest neighbor).

5 Information from Coreference and Relations

Our system is processing a *document* consisting of multiple sentences. For each sentence, the name recognizer generates multiple hypotheses, each of which is an NE tagging of the entire sentence. The names in the hypothesis, plus the nouns in the categories of interest constitute the mention set for that hypothesis. Coreference resolution links these mentions, assigning each to an entity. In symbols:

S_i is the i-th sentence in the document.

H_i is the hypotheses set for S_i

h_{ij} is the j-th hypothesis in S_i

M_{ij} is the mention set for h_{ij}

m_{ijk} is the k-th mention in M_{ij}

e_{ijk} is the entity which m_{ijk} belongs to according to the current reference resolution results

5.1 Coreference Features

For each mention we compute seven quantities based on the results of name tagging and reference resolution:

[1] This constraint is relaxed for parallel structures such as "mention1, mention2, [and] mention3…"; in such cases there can be more than one intervening mention.

$CorefNum_{ijk}$ is the number of mentions in e_{ijk}

$WeightSum_{ijk}$ is the sum of all the link weights between m_{ijk} and other mentions in e_{ijk}, 0.8 for name-name coreference; 0.5 for apposition; 0.3 for other name-nominal coreference

$FirstMention_{ijk}$ is 1 if m_{ijk} is the first name mention in the entity; otherwise 0

$Head_{ijk}$ is 1 if m_{ijk} includes the head word of name; otherwise 0

$Withoutidiom_{ijk}$ is 1 if m_{ijk} is not part of an idiom; otherwise 0

$PERContext_{ijk}$ is the number of PER context words around a PER name such as a title or an action verb involving a PER

$ORGSuffix_{ijk}$ is 1 if ORG m_{ijk} includes a suffix word; otherwise 0

The first three capture evidence of the correctness of a name provided by reference resolution; for example, a name which is coreferenced with more other mentions is more likely to be correct. The last four capture local or name-internal evidence; for instance, that an organization name includes an explicit, organization-indicating suffix.

We then compute, for each of these seven quantities, the sum over all mentions k in a sentence, obtaining values for $CorefNum_{ij}$, $WeightSum_{ij}$, etc.:

$$CorefNum_{ij} = \sum_k CorefNum_{ijk} \quad \text{etc.}$$

Finally, we determine, for a given sentence and hypothesis, for each of these seven quantities, whether this quantity achieves the maximum of its values for this hypothesis:

$BestCorefNum_{ij} \equiv$

$CorefNum_{ij} = \max_q CorefNum_{iq}$ etc.

We will use these properties of the hypothesis as features in assessing the quality of a hypothesis.

5.2 Relation Word Clusters

In addition to using relation information for reranking name hypotheses, we used the relation training corpus to build word clusters which could more directly improve name tagging. Name taggers rely heavily on words in the immediate context to identify and classify names; for example, specific job titles, occupations, or family relations can be used to identify people names. Such words are learned individually from the name tagger's training corpus. If we can provide the name tagger with clusters of related words, the tagger will be able to generalize from the examples in the training corpus to other words in the cluster.

The set of ACE relations includes several involving employment, social, and family relations. We gathered the words appearing as an argument of one of these relations in the training corpus, eliminated low-frequency terms and manually edited the ten resulting clusters to remove inappropriate terms. These were then combined with lists (of titles, organization name suffixes, location suffixes) used in the baseline tagger.

5.3 Relation Features

Because the performance of our relation tagger is not as good as our coreference resolver, we have used the results of relation detection in a relatively simple way to enhance name detection. The basic intuition is that a name which has been correctly identified is more likely to participate in a relation than one which has been erroneously identified.

For a given range of margins (from the HMM), the probability that a name in the first hypothesis is correct is shown in the following table, for names participating and not participating in a relation:

Margin	In Relation(%)	Not in Relation(%)
<4	90.7	55.3
<3	89.0	50.1
<2	86.9	42.2
<1.5	81.3	28.9
<1.2	78.8	23.1
<1	75.7	19.0
<0.5	66.5	14.3

Table 1 Probability of a name being correct

Table 1 confirms that names participating in relations are much more likely to be correct than names that do not participate in relations. We also see, not surprisingly, that these probabilities are strongly affected by the HMM hypothesis margin (the difference in log probabilities) between the first hypothesis and the second hypothesis. So it is natural to use participation in a relation (coupled with a margin value) as a valuable feature for reranking name hypotheses.

Let m_{ijk} be the k-th name mention for hypothesis h_{ij} of sentence; then we define:

$Inrelation_{ijk}$ = 1 if m_{ijk} is in a definite relation
= 0 if m_{ijk} is in a possible relation
= -1 if m_{ijk} is not in a relation

$$Inrelation_{ij} = \sum_k Inrelation_{ijk}$$

$$Mostrelated_{ij} \equiv (Inrelation_{ij} = \max_q Inrelation_{iq})$$

Finally, to capture the interaction with the margin, we let z_i = the margin for sentence S_i and divide the range of values of z_i into six intervals Mar_1, … Mar_6. And we define the hypothesis ranking information: $FirstHypothesis_{ij}$ = 1 if j =1; otherwise 0.

We will use as features for ranking h_{ij} the conjunction of $Mostrelated_{ij}$, $z_i \in Mar_p$ (p = 1, …, 6), and $FirstHypothesis_{ij}$.

6 Using the Information from Coreference and Relations

6.1 Word Clustering based on Relations

As we described in section 5.2, we can generate word clusters based on relation information. If a word is not part of a relation cluster, we consider it an independent (1-word) cluster.

The Nymble name tagger (Bikel et al., 1999) relies on a multi-level linear interpolation model for backoff. We extended this model by adding a level from word to cluster, so as to estimate more reliable probabilities for words in these clusters. Table 2 shows the extended backoff model for each of the three probabilities used by Nymble.

Transition Probability	First-Word Emission Probability	Non-First-Word Emission Probability
$P(NC_2\|NC_1, <w_1, f_1>)$	$P(<w_2,f_2>\|$ $NC_1, NC_2)$	$P(<w_2,f_2>\|$ $<w_1,f_1>, NC_2)$
	$P(<Cluster_2,f_2>\|$ $NC_1, NC_2)$	$P(<Cluster_2,f_2>\|$ $<w_1,f_1>, NC_2)$
$P(NC_2\|NC_1,$ $<Cluster_1,$ $f_1>)$	$P(<Cluster_2,f_2>\|$ <+begin+, other>, $NC_2)$	$P(<Cluster_2,f_2>\|$ $<Cluster_1,f_1>,$ $NC_2)$
$P(NC_2\|NC_1)$	$P(<Cluster_2, f_2>\|NC_2)$	
$P(NC_2)$	$P(Cluster_2\|NC_2) * P(f_2\|NC_2)$	
1/#(name classes)	1/#(cluster) * 1/#(word features)	

Table2 Extended Backoff Model

6.2 Pre-pruning by Margin

The HMM tagger produces the N best hypotheses for each sentence.[2] In order to decide when we need to rely on global (coreference and relation) information for name tagging, we want to have some assessment of the confidence that the name tagger has in the first hypothesis. In this paper, we use the margin for this purpose. A large margin indicates greater confidence that the first hypothesis is correct.[3] So if the margin of a sentence is above a threshold, we select the first hypothesis, dropping the others and by-passing the reranking.

6.3 Re-ranking based on Coreference

We described in section 5.1, above, the coreference features which will be used for reranking the hypotheses after pre-pruning. A maximum entropy model for re-ranking these hypotheses is then trained and applied as follows:

Training

1. Use K-fold cross-validation to generate multiple name tagging hypotheses for each document in the training data D_{train} (in each of the K iterations, we use K-1 subsets to train the HMM and then generate hypotheses from the K^{th} subset).
2. For each document d in D_{train}, where d includes n sentences $S_1 \ldots S_n$
   For i = 1…n, let m = the number of hypotheses for S_i
   (1) Pre-prune the candidate hypotheses using the HMM margin
   (2) For each hypothesis h_{ij}, j = 1…m
      (a) Compare h_{ij} with the key, set the prediction $Value_{ij}$ "Best" or "Not Best"
      (b) Run the Coreference Resolver on h_{ij} and the best hypothesis for each of the other sentences, generate entity results for each candidate name in h_{ij}
      (c) Generate a coreference feature vector V_{ij} for h_{ij}
      (d) Output V_{ij} and $Value_{ij}$

[2] We set different N = 5, 10, 20 or 30 for different margin ranges, by cross-validation checking the training data about the ranking position of the best hypothesis for each sentence. With this N, optimal reranking (selecting the best hypothesis among the N best) would yield Precision = 96.9 Recall = 94.5 F = 95.7 on our test corpus.

[3] Similar methods based on HMM margins were used by (Scheffer et al., 2001).

3. Train Maxent Re-ranking system on all V_{ij} and $Value_{ij}$

Test

1. Run the baseline name tagger to generate multiple name tagging hypotheses for each document in the test data D_{test}
2. For each document d in D_{test}, where d includes n sentences $S_1 \ldots S_n$
   (1) Initialize: Dynamic input of coreference resolver H = {$h_{i\text{-}best}$ | i = 1…n, $h_{i\text{-}best}$ is the current best hypothesis for S_i}
   (2) For i = 1…n, assume m = the number of hypotheses for S_i
      (a) Pre-prune the candidate hypotheses using the HMM margin
      (b) For each hypothesis h_{ij}, j = 1…m
         - $h_{i\text{-}best} = h_{ij}$
         - Run the Coreference Resolver on H, generate entity results for each name candidate in h_{ij}
         - Generate a coreference feature vector V_{ij} for h_{ij}
         - Run Maxent Re-ranking system on V_{ij}, produce $Prob_{ij}$ of "Best" value
      (c) $h_{i\text{-}best}$ = the hypothesis with highest $Prob_{ij}$ of "Best" value, update H and output $h_{i\text{-}best}$

6.4 Re-ranking based on Relations

From the above first-stage re-ranking by coreference, for each hypothesis we got the probability of its being the best one. By using these results and relation information we proceed to a second-stage re-ranking. As we described in section 5.3, the information of "in relation or not" can be used together with margin as another important measure of confidence.

In addition, we apply the mechanism of weighted voting among hypotheses (Zhai et al., 2004) as an additional feature in this second-stage re-ranking. This approach allows all hypotheses to vote on a possible name output. A recognized name is considered correct only when it occurs in more than 30 percent of the hypotheses (weighted by their probability).

In our experiments we use the probability produced by the HMM, $prob_{ij}$, for hypothesis h_{ij}. We normalize this probability weight as:

$$W_{ij} = \frac{\exp(prob_{ij})}{\sum_q \exp(prob_{iq})}$$

For each name mention m_{ijk} in h_{ij}, we define:

$Occur_q(m_{ijk})$ = 1 if m_{ijk} occurs in h_q

= 0 otherwise

Then we count its voting value as follows:

$Voting_{ijk}$ is 1 if $\sum_q W_{iq} \times Occur_q(m_{ijk}) > 0.3$;

otherwise 0.

The voting value of h_{ij} is:

$$Voting_{ij} = \sum_k Voting_{ijk}$$

Finally we define the following voting feature:

$BestVoting_{ij} \equiv (Voting_{ij} = \max_q Voting_{iq})$

This feature is used, together with the features described at the end of section 5.3 and the probability score from the first stage, for the second-stage maxent re-ranking model.

One appeal of the above two re-ranking algorithms is its flexibility in incorporating features into a learning model: essentially any coreference or relation features which might be useful in discriminating good from bad structures can be included.

7 System Pipeline

Combining all the methods presented above, the flow of our final system is shown in figure 1.

8 Evaluation Results

8.1 Training and Test Data

We took 346 documents from the 2004 ACE training corpus and official test set, including both broadcast news and newswire, as our blind test set. To train our name tagger, we used the Beijing University Insititute of Computational Linguistics corpus – 2978 documents from the People's Daily in 1998 – and 667 texts in the training corpus for the 2003 & 2004 ACE evaluation. Our reference resolver is trained on these 667 ACE texts. The relation tagger is trained on 546 ACE 2004 texts, from which we also extracted the relation clusters. The test set included 11715 names: 3551 persons, 5100 GPEs and 3064 organizations.

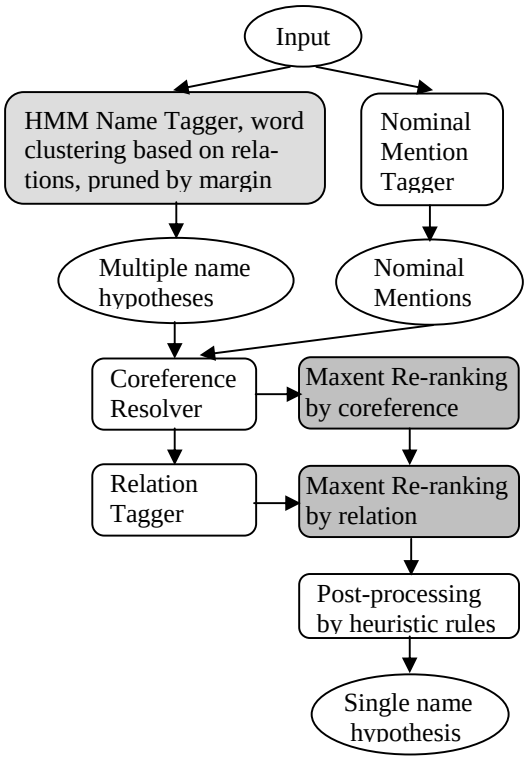

Figure 1 System Flow

8.2 Overall Performance Comparison

Table 3 shows the performance of the baseline system; Table 4 is the system with relation word clusters; Table 5 is the system with both relation clusters and re-ranking based on coreference features; and Table 6 is the whole system with second-stage re-ranking using relations.

The results indicate that relation word clusters help to improve the precision and recall of most name types. Although the overall gain in F-score is small (0.7%), we believe further gain can be achieved if the relation corpus is enlarged in the future. The re-ranking using the coreference features had the largest impact, improving precision and recall consistently for all types. Compared to our system in (Ji and Grishman, 2004), it helps to distinguish the good and bad hypotheses without any loss of recall. The second-stage re-ranking using the relation participation feature yielded a small further gain in F score for each type, improving precision at a slight cost in recall.

The overall system achieves a 24.1% relative reduction on the spurious and incorrect tags, and 14.3% reduction in the missing rate over a state-of-the-art baseline HMM trained on the same material. Furthermore, it helps to disambiguate many name type errors: the number of cases of type confusion in name classification was reduced from 191 to 102.

Name	Precision	Recall	F
PER	88.6	89.2	88.9
GPE	88.1	84.9	86.5
ORG	88.8	87.3	88.0
ALL	88.4	86.7	87.5

Table 3 Baseline Name Tagger

Name	Precision	Recall	F
PER	89.4	90.1	89.7
GPE	88.9	85.8	89.4
ORG	88.7	87.4	88.0
ALL	89.0	87.4	88.2

Table 4 Baseline + Word Clustering by Relation

Name	Precision	Recall	F
PER	90.1	91.2	90.5
GPE	89.7	86.8	88.2
ORG	90.6	89.8	90.2
ALL	90.0	88.8	89.4

Table 5 Baseline + Word Clustering by Relation + Re-ranking by Coreference

Name	Precision	Recall	F
PER	90.7	91.0	90.8
GPE	91.2	86.9	89.0
ORG	91.7	89.1	90.4
ALL	91.2	88.6	89.9

Table 6 Baseline + Word Clustering by Relation + Re-ranking by Coreference + Re-ranking by Relation

In order to check how robust these methods are, we conducted significance testing (sign test) on the 346 documents. We split them into 5 folders, 70 documents in each of the first four folders and 66 in the fifth folder. We found that each enhancement (word clusters, coreference reranking, relation reranking) produced an improvement in F score for each folder, allowing us to reject the hypothesis that these improvements were random at a 95% confidence level. The overall F-measure improvements (using all enhancements) for the 5 folders were: 2.3%, 1.6%, 2.1%, 3.5%, and 2.1%.

9 Conclusion

This paper explored methods for exploiting the interaction of analysis components in an information extraction system to reduce the error rate of individual components. The ACE task hierarchy provided a good opportunity to explore these interactions, including the one presented here between reference resolution/relation detection and name tagging. We demonstrated its effectiveness for Chinese name tagging, obtaining an absolute improvement of 2.4% in F-measure (a reduction of 19% in the (1 – F) error rate). These methods are quite low-cost because we don't need any extra resources or components compared to the baseline information extraction system.

Because no language-specific rules are involved and no additional training resources are required, we expect that the approach described here can be straightforwardly applied to other languages. It should also be possible to extend this re-ranking framework to other levels of analysis in information extraction –- for example, to use event detection to improve name tagging; to incorporate subtype tagging results to improve name tagging; and to combine name tagging, reference resolution and relation detection to improve nominal mention tagging. For Chinese (and other languages without overt word segmentation) it could also be extended to do character-based name tagging, keeping multiple segmentations among the N-Best hypotheses. Also, as information extraction is extended to capture cross-document information, we should expect further improvements in performance of the earlier stages of analysis, including in particular name identification.

For some levels of analysis, such as name tagging, it will be natural to apply lattice techniques to organize the multiple hypotheses, at some gain in efficiency.

Acknowledgements

This research was supported by the Defense Advanced Research Projects Agency under Grant N66001-04-1-8920 from SPAWAR San Diego, and by the National Science Foundation under Grant 03-25657. This paper does not necessarily reflect the position or the policy of the U.S. Government.

References

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. *Nymble: a high-performance Learning Name-finder*. Proc. Fifth Conf. on Applied Natural Language Processing, Washington, D.C.

Andrew Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. Dissertation, Dept. of Computer Science, New York University.

Hai Leong Chieu and Hwee Tou Ng. 2002. *Named Entity Recognition: A Maximum Entropy Approach Using Global Information.* Proc.: 17th Int'l Conf. on Computational Linguistics (COLING 2002), Taipei, Taiwan.

Yen-Lu Chow and Richard Schwartz. 1989. *The N-Best Algorithm: An efficient Procedure for Finding Top N Sentence Hypotheses.* Proc. DARPA Speech and Natural Language Workshop

Michael Collins. 2002. *Ranking Algorithms for Named-Entity Extraction: Boosting and the Voted Perceptron*. Proc. ACL 2002

Heng Ji and Ralph Grishman. 2004. *Applying Coreference to Improve Name Recognition.* Proc. ACL 2004 Workshop on Reference Resolution and Its Applications, Barcelona, Spain

N. Kambhatla. 2004. *Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations*. Proc. ACL 2004.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. *Active Hidden Markov Models for Information Extraction*. Proc. Int'l Symposium on Intelligent Data Analysis (IDA-2001).

Dmitry Zelenko, Chinatsu Aone, and Jason Tibbets. 2004. *Binary Integer Programming for Information Extraction*. ACE Evaluation Meeting, September 2004, Alexandria, VA.

Lufeng Zhai, Pascale Fung, Richard Schwartz, Marine Carpuat, and Dekai Wu. 2004. *Using N-best Lists for Named Entity Recognition from Chinese Speech*. Proc. NAACL 2004 (Short Papers)