Analysis and Repair of Name Tagger Errors

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

Name tagging is a critical early stage in many natural language processing pipelines. In this paper we analyze the types of errors produced by a tagger, distinguishing name classification and various types of name identification errors. We present a joint inference model to improve Chinese name tagging by incorporating feedback from subsequent stages in an information extraction pipeline: name structure parsing, cross-document coreference, semantic relation extraction and event extraction. We show through examples and performance measurement how different stages can correct different types of errors. The resulting accuracy approaches that of individual human annotators.

1 Introduction

High-performance named entity (NE) tagging is crucial in many natural language processing tasks, such as information extraction and machine translation. In 'traditional' pipelined system architectures, NE tagging is one of the first steps in the pipeline. NE errors adversely affect subsequent stages, and error rates are often *compounded* by later stages.

However, (Roth and Yi 2002, 2004) and our recent work have focused on incorporating richer linguistic analysis, using the "feedback" from later stages to improve name taggers. We expanded our last year's model (Ji and Grishman, 2005) that used the results of coreference analysis and relation extraction, by adding 'feedback' from more information extraction components – name structure parsing, cross-document coreference, and event extraction – to incrementally rerank the multiple hypotheses from a baseline name tagger.

While together these components produced a further improvement on last year's model, our goal in this paper is to look behind the overall performance figures in order to understand how these varied components contribute to the improvement, and compare the remaining system errors with the human annotator's performance. To this end, we shall decompose the task of name tagging into two subtasks

- Name Identification – The process of identifying name boundaries in the sentence.
- Name Classification – Given the correct name boundaries, assigning the appropriate name types to them.

and observe the effects that different components have on errors of each type. Errors of identification will be further subdivided by type (missing names, spurious names, and boundary errors). We believe such detailed understanding of the benefits of joint inference is a prerequisite for further improvements in name tagging performance.

After summarizing some prior work in this area, describing our baseline NE tagger, and analyzing its errors, we shall illustrate, through a series of examples, the potential for feedback to improve NE performance. We then present some details on how this improvement can be achieved through hypothesis reranking in the extraction pipeline, and analyze the results in terms of different types of identification and classification errors.

2 Prior Work

Some recent work has incorporated global information to improve the performance of name taggers.

For mixed case English data, name identification is relatively easy. Thus some researchers have focused on the more challenging task – classifying names into correct types. In (Roth and

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 420–427,
Sydney, July 2006. ©2006 Association for Computational Linguistics

Yi 2002, 2004), given name boundaries in the text, separate classifiers are first trained for name classification and semantic relation detection. Then, the output of the classifiers is used as a conditional distribution given the observed data. This information, along with the constraints among the relations and entities (specific relations require specific classes of names), is used to make global inferences by linear programming for the most probable assignment. They obtained significant improvements in both name classification and relation detection.

In (Ji and Grishman 2005) we generated N-best NE hypotheses and re-ranked them after coreference and semantic relation identification; we obtained a significant improvement in Chinese name tagging performance. In this paper we shall use a wider range of linguistic knowledge sources, and integrate cross-document techniques.

3 Baseline Name Tagger

We apply a multi-lingual (English / Chinese) bigram HMM tagger to identify four named entity types: Person, Organization, GPE (‘geo-political entities’ – locations which are also political units, such as countries, counties, and cities) and Location. The HMM tagger generally follows the Nymble model (Bikel et al, 1997), and uses best-first search to generate N-Best hypotheses for each input sentence.

In mixed-case English texts, most proper names are capitalized. So capitalization provides a crucial clue for name boundaries.

In contrast, a Chinese sentence is composed of a string of characters without any word boundaries or capitalization. Even after word segmentation there are still no obvious clues for the name boundaries. However, we can apply the following coarse “usable-character” restrictions to reduce the search space.

Standard Chinese family names are generally single characters drawn from a set of 437 family names (there are also 9 two-character family names, although they are quite infrequent) and given names can be one or two characters (Gao et al., 2005). Transliterated Chinese person names usually consist of characters in three relatively fixed character lists (Begin character list, Middle character list and End character list). Person abbreviation names and names including title words match a few patterns. The suffix words (if there are any) of Organization and GPE names belong to relatively fixed lists too.

However, this “usable-character” restriction is not as reliable as the capitalization information for English, since each of these special characters can also be part of common words.

3.1 Identification and Classification Errors

We begin our error analysis with an investigation of the English and Chinese baseline taggers, decomposing the errors into identification and classification errors. In Figure 1 we report the identification F-Measure for the baseline (the first hypothesis), and the N-best upper bound, the best of the N hypotheses[1], using different models: English MonoCase (EN-Mono, without capitalization), English Mixed Case (EN-Mix, with capitalization), Chinese without the usable character restriction (CH-NoRes) and Chinese with the usable character restriction (CH-WithRes).

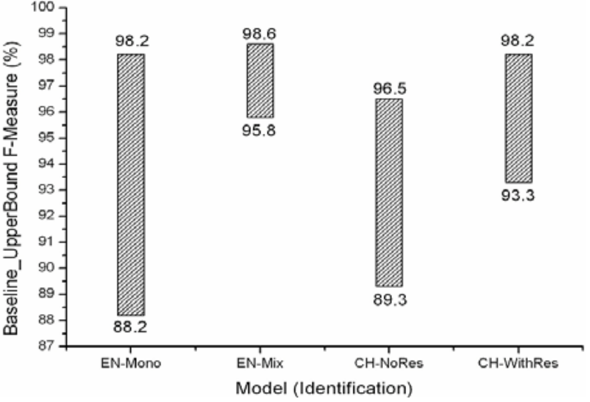

Figure 1. Baseline and Upper Bound of Name Identification

Figure 1 shows that capitalization is a crucial clue in English name identification (increasing the F measure by 7.6% over the monocase score). We can also see that the best of the top N (N <= 30) hypotheses is very good, so reranking a small number of hypotheses has the potential of producing a very good tagger.

The “usable” character restriction plays a major role in Chinese name identification, increasing the F-measure 4%. With this restriction, the performance of the best-of-N-best is again very good. However, it is evident that, even with this restriction, identification is more challenging for Chinese, due to the absence of capitalization and word boundaries.

Figure 2 shows the classification accuracy of the above four models. We can see that capitalization does not help English name classification;

[1] These figures were obtained using training and test corpora described later in this paper, and a value of N ranging from 1 to 30 depending on the margin of the HMM tagger, as also described below. All figures are with respect to the official ACE keys prepared by the Linguistic Data Consortium.

and the difficulty of classification is similar for the two languages.

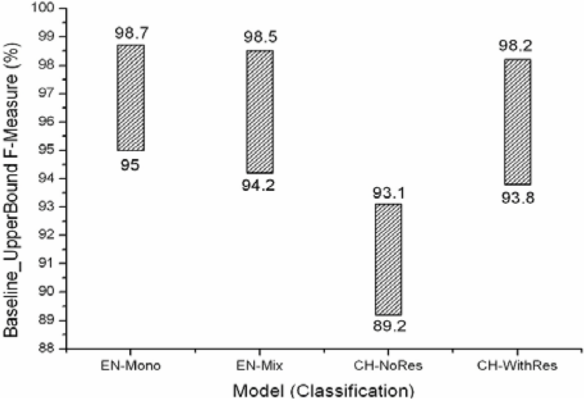

Figure 2. Baseline and Upper Bound of Name Classification

3.2 Identification Errors in Chinese

For the remainder of this paper we shall focus on the more difficult problems of Chinese tagging, using the HMM system with character restrictions as our baseline. The name identification errors of this system can be divided into missed names (21%), spurious names (29%), and boundary errors, where there is a partial overlap between the names in the key and the system response (50%). Confusion between names and nominals (phrases headed by a common noun) is a major source of both missed and spurious names (56% of missed, 24% of spurious). In a language without capitalization, this is a hard task even for people; one must rely largely on world knowledge to decide whether a phrase (such as the "criminal-processing team") is an organization name or merely a description of an organization. The other major source of missed names is words not seen in the training data, generally representing minor cities or other locations in China (28%). For spurious names, the largest source of error is names of a type not included in the key (44%) which are mistakenly tagged as one of the known name types.[2] As we shall see, different types of knowledge are required for correcting different types of errors.

4 Mutual Inferences between Information Extraction Stages

4.1 Extraction Pipeline

Name tagging is typically one of the first stages in an information extraction pipeline. Specifically, we will consider a system which was developed for the ACE (Automatic Content Extraction) task[3] and includes the following stages: name structure parsing, coreference, semantic relation extraction and event extraction (Ji et al., 2006).

All these stages are performed after name tagging since they take names as input “objects”. However, the inferences from these subsequent stages can also provide valuable constraints to identify and classify names.

Each of these stages connects the name candidate to other linguistic elements in the sentence, document, or corpus, as shown in Figure 3.

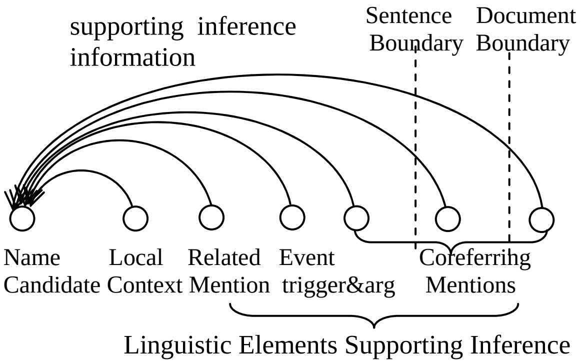

Figure 3. Name candidate and its global context

The baseline name tagger (HMM) uses very local information; feedback from later extraction stages allows us to draw from a wider context in making final name tagging decisions.

In the following we use two related (translated) texts as examples, to give some intuition of how these different types of linguistic evidence improve name tagging.[4]

Document 1: Yugoslav election

> […] More than 300,000 people rushed the *<bei er ge le>*$_0$ congress building, forcing *<yugoslav>*$_1$ president *<mi lo se vi c>*$_2$ to admit frankly that in the Sept. 24 election he was beaten by his opponent *<ke shi tu ni cha>*$_3$.
>
> *<mi lo se vi c>*$_4$ was forced to flee *<bei er ge le>*$_5$; the winning opposition party's *<sai er wei ya>*$_6$ *<anti-democracy committee>*$_7$ on the morning of the 6th formed a *<crisis-handling*

[2] If the key included an 'other' class of names, these would be classification errors; since it does not -- since these names are not tagged in the key -- the automatic scorer treats them as spurious names.

[3] The ACE task description can be found at http://www.itl.nist.gov/iad/894.01/tests/ace/ and the ACE guidelines at http://www.ldc.upenn.edu/Projects/ACE/

[4] Rather than offer the most fluent translation, we have provided one that more closely corresponds to the Chinese text in order to more clearly illustrate the linguistic issues. Transliterated names are rendered phonetically, character by character.

committee>$_8$, to deal with transfer-of-power issues.

This crisis committee includes police, supply, economics and other important departments.

In such a crisis, people cannot think through this question: has the <*yugoslav*>$_9$ president <*mi lo se vi c*>$_{10}$ used up his skills?

According to the official voting results in the first round of elections, <*mi lo se vi c*>$_{11}$ was beaten by *<18 party opposition committee>*$_{12}$ candidate <*ke shi tu ni cha*>$_{13}$. […]

Document 2: Biography of these two leaders

[…]<*ke shi tu ni cha*>$_{14}$ used to pursue an academic career, until 1974, when due to his opposition position he was fired by <*bei er ge le*>$_{15}$ <*law school*>$_{16}$ and left the academic community. <*ke shi tu ni cha*>$_{17}$ also at the beginning of the 1990s joined the opposition activity, and in 1992 founded <*sai er wei ya*>$_{18}$ <*opposition party*>$_{19}$.

This famous new leader and his previous classmate at law school, namely his wife <*zuo li ka*>$_{20}$ live in an apartment in <*bei er ge le*>$_{21}$.

The vanished <*mi lo se vi c*>$_{22}$ was born in <*sai er wei ya*>$_{23}$ ‘s central industrial city. […]

4.1 Inferences for Correcting Name Errors

4.2.1 Internal Name Structure

Constraints and preferences on the structure of individual names can capture local information missed by the baseline name tagger. They can correct several types of identification errors, including in particular boundary errors. For example, “<*ke shi tu ni cha*>$_3$” is more likely to be correct than “<*shi tu ni cha*>$_3$” since “*shi*” (什) cannot be the first character of a transliterated name.

Name structures help to classify names too. For example, “*anti-democracy committee*$_7$” is parsed as “[Org-Modifier anti-democracy] [Org-Suffix committee]”, and the first character is not a person last name or the first character of a transliterated person name, so it is more likely to be an organization than a person name.

4.2.2 Patterns

Information about expected sequences of constituents surrounding a name can be used to correct name *boundary errors*. In particular, event extraction is performed by matching patterns involving a "trigger word" (typically, the main verb or nominalization representing the event) and a set of arguments. When a name candidate is involved in an event, the trigger word and other arguments of the event can help to determine the name boundaries. For example, in the sentence “*The vanished mi lo se vi c was born in sai er wei ya ‘s central industrial city*”, “*mi lo se vi c*” is more likely to be a name than “*mi lo se*”, “*sai er wei ya”* is more likely be a name than *“er wei”*, because these boundaries will allow us to match the event pattern “[Adj] [PER-NAME] [Trigger word for 'born' event] in [GPE-NAME]’s [GPE-Nominal]”.

4.2.3 Selection

Any context which can provide selectional constraints or preferences for a name can be used to correct name *classification* errors. Both semantic relations and events carry selectional constraints and so can be used in this way.

For instance, if the “*Personal-Social/Business*” relation (“*opponent*”) between “*his*” and “*<ke shi tu ni cha>*$_3$” is correctly identified, it can help to classify “*<ke shi tu ni cha>*$_3$” as a person name. Relation information is sometimes crucial to classifying names. “<*mi lo se vi c*>$_{10}$” and “<*ke shi tu ni cha*>$_{13}$” are likely person names because they are “*employees*” of “<*yugoslav*>$_9$” and “*<18 party opponent committee>*$_{12}$”. Also the “*Personal-Social/Family*” relation (“*wife*”) between *“his*” and “<*zuo li ka*>$_{20}$” helps to classify <*zuo li ka*>$_{20}$ as a person name.

Events, like relations, can provide effective selectional preferences to correctly classify names. For example, “<*mi lo se vi c*>$_{2,4,10,11,22}$” are likely person names because they are involved in the following events: “*claim*”, “*escape*”, “*built*”, “*beat*”, “*born*”, while “<*sai er wei ya*>$_{23}$”can be easily tagged as GPE because it’s a “*birth-place*” in the event “*born*”.

4.2.4 Coreference

Names which are introduced in an article are likely to be referred to again, either by repeating the same name or describing it with nominal mentions (phrases headed by common nouns). These mentions will have the same spelling (though if a name has several parts, some may be dropped) and same semantic type. So if the boundary or type of one mention can be determined with some confidence, coreference can be used to disambiguate other mentions.

For example, if “< *mi lo se vi c*>$_2$” is confirmed as a name, then “< *mi lo se vi c*>$_{10}$” is more likely to be a name than “< *mi lo se*>$_{10}$”, by refering to "< *mi lo se vi c*>$_{2}$". Also "This crisis committee" supports the analysis of "<*crisis-handling committee*>$_{8}$" as an organization name in preference to the alternative name candidate "<*crisis-handling*>$_{8}$".

For a name candidate, high-confidence information about the type of one mention can be used to determine the type of other mentions. For example, for the repeated person name "< *mi lo se vi c*>$_{2,4,10,11,22}$" type information based on the event context of one mention can be used to classify or confirm the type of the others. The person nominal "This famous new leader" confirms "<*ke shi tu ni cha*>$_{17}$" as a person name.

5 Incremental Re-Ranking Algorithm

5.1 Overall Architecture

In this section we will present the algorithms to capture the intuitions described in Section 4. The overall system pipeline is presented in Figure 4.

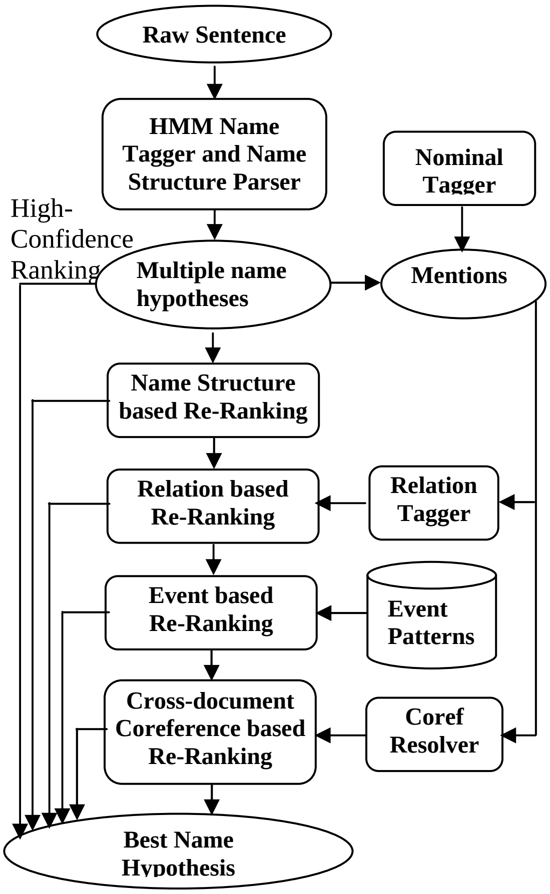

Figure 4. System Architecture

The baseline name tagger generates N-Best multiple hypotheses for each sentence, and 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. A large margin indicates greater confidence that the first hypothesis is correct.[5] It generates name structure parsing results too, such as the family name and given name of person, the prefixes of the abbreviation names, the modifiers and suffixes of organization names.

Then the results from subsequent components are exploited in four incremental re-rankers. From each re-ranking step we output the best name hypothesis directly if the re-ranker has high confidence in its decisions. Otherwise the sentence is forwarded to the next re-ranker, based on other features. In this way we can adjust the ranking of multiple hypotheses and select the best tagging for each sentence gradually.

The nominal mention tagger (noun phrase chunker) uses a maximum entropy model. Entity type assignment for the nominal heads is done by table look-up. The coreference resolver is a combination of high-precision heuristic rules and maximum entropy models. In order to incorporate wider context we use cross-document coreference for the test set. We cluster the documents using a cross-entropy metric and then treat the entire cluster as a single document.

The relation tagger uses a K-nearest-neighbor algorithm.

We extract event patterns from the ACE05 training corpus for personnel, contact, life, business, and conflict events. We also collect additional event trigger words that appear frequently in name contexts, from a syntactic dictionary, a synonym dictionary and Chinese PropBank V1.0. Then the patterns are generalized and tested semi-automatically.

5.2 Supervised Re-Ranking Model

In our name re-ranking model, each hypothesis is an NE tagging of *the entire sentence*, for example, "*The vanished <PER>mi lo se vi c</PER> was born in <GPE>sai er wei ya</GPE>'s central industrial city*"; and each pair of hypotheses (h_i, h_j) is called a "sample".

[5] The margin also determines the number of hypotheses (N) generated by the baseline tagger. Using cross-validation on the training data, we determine the value of N required to include the best hypothesis, as a function of the margin. We then divide the margin into ranges of values, and set a value of N for each range, with a maximum of 30.

Re-Ranker	Property for comparing names N_{ik} and N_{jk}	
Name Structure Based	*HMMMargin*	scaled margin value from HMM
	Idiom$_{ik}$	-1 if N_{ik} is part of an idiom; otherwise 0
	PERContext$_{ik}$	the number of PER context words if N_{ik} and N_{jk} are both PER; otherwise 0
	ORGSuffix$_{ik}$	1 if N_{ik} is tagged as ORG and it includes a suffix word; otherwise 0
	PERCharacter$_{ik}$	-1 if N_{ik} is tagged as PER without family name, and it does not consist entirely of transliterated person name characters; otherwise 0
	Titlestructure$_{ik}$	-1 if N_{ik} = title word + family name while N_{jk} = title word + family name + given name; otherwise 0
	Digit$_{ik}$	-1 if N_{ik} is PER or GPE and it includes digits or punctuation; otherwise 0
	AbbPER$_{ik}$	-1 if N_{ik} = little/old + family name + given name while N_{jk} = little/old + family name; otherwise 0
	SegmentPER$_{ik}$	-1 if N_{ik} is GPE (PER)* GPE , while N_{jk} is PER*; otherwise 0
	Voting$_{ik}$	the voting rate among all the candidate hypotheses[6]
	FamousName$_{ik}$	1 if N_{ik} is tagged as the same type in one of the famous name lists[7]; otherwise 0
Relation Based	*Probability1$_i$*	scaled ranking probability for (h_i, h_j) from name structure based re-ranker
	Relation Constraint$_{ik}$	If N_{ik} is in relation R (N_{ik} = EntityType$_1$, M$_2$ = EntityType$_2$), compute Prob(EntityType$_1$|EntityType$_2$, R) from training data and scale it; otherwise 0
	Conjunction of InRelation$_i$ & Probability1$_i$	*Inrelation$_{ik}$* is 1 if N_{ik} and N_{jk} have different name types, and N_{ik} is in a definite relation while N_{jk} is not; otherwise 0. $Inrelation_i = \sum_k Inrelation_{ik}$
Event Based	*Probability2$_i$*	scaled ranking probability for (h_i, h_j) from relation based re-ranker
	Event Constraint$_i$	1 if all entity types in h_i match event pattern, -1 if some do not match, and 0 if the argument slots are empty
	EventSubType	Event subtype if the patterns are extracted from ACE data, otherwise"None"
Cross-document Coreference Based	*Probability3$_i$*	scaled ranking probability for (h_i, h_j) from event based re-ranker
	Head$_{ik}$	1 if N_{ik} includes the head word of name; otherwise 0
	CorefNum$_{ik}$	the number of mentions corefered to N_{ik}
	WeightNum$_{ik}$	the sum of all link weights between N_{ik} and its corefered mentions, 0.8 for name-name coreference; 0.5 for apposition; 0.3 for other name-nominal coreference
	NumHighCoref$_i$	the number of mentions which corefer to N_{ik} and output by previous re-rankers with high confidence

Table 3. Re-Ranking Properties

Component		Data
Training	Baseline name tagger	2978 texts from the People's Daily in 1998 and 1300 texts from ACE03, 04, 05 training data
	Nominal tagger	Chinese Penn TreeBank V5.1
	Coreference resolver	1300 texts from ACE03, 04, 05 training data
	Relation tagger	633 ACE 05 texts, and 546 ACE 04 texts with types/subtypes mapped into 05 set
	Event pattern	376 trigger words, 661 patterns
	Name structure, coreference and relation based re-rankers	1,071,285 samples (pairs of hypotheses) from ACE 03, 04 and 05 training data
	Event based re-ranker	325,126 samples from ACE sentences including event trigger words
Test		100 texts from ACE 04 training corpus, includes 2813 names: 1126 persons, 712 GPEs, 785 organizations and 190 locations.

Table 4. Data Description

[6] The method of counting the voting rate refers to (Zhai, 04) and (Ji and Grishman, 05)

[7] Extracted from the high-frequency name lists from the training corpus, and country/province/state/ city lists from Chinese wikipedia.

The goal of each re-ranker is to learn a ranking function f of the following form: for each pair of hypotheses (h_i, h_j), f : H × H → {-1, 1}, such that f(h_i, h_j) = 1 if h_i is better than h_j; f (h_i, h_j) = -1 if h_i is worse than h_j. In this way we are able to convert ranking into a classification problem. And then a maximum entropy model for re-ranking these hypotheses can be trained and applied.

During training we use F-measure to measure the quality of each name hypothesis against the key. During test we get from the MaxEnt classifier the probability (ranking confidence) for each pair: Prob (f (h_i, h_j) = 1). Then we apply a dynamic decoding algorithm to output the best hypothesis. More details about the re-ranking algorithm are presented in (Ji et al., 2006).

5.3 Re-Ranking Features

For each sample (h_i, h_j), we construct a feature set for assessing the ranking of h_i and h_j. Based on the information obtained from inferences, we compute (for each property) the property score PS_{ik} for each individual name candidate N_{ik} in h_i; some of these properties depend also on the corresponding name tags in h_j. Then we sum over all names in each hypothesis h_i:

$$PS_i = \sum_k PS_{ik}$$

Finally we use the quantity $(PS_i – PS_j)$ as the feature value for the sample (h_i, h_j). Table 3 summarizes the property scores PS_{ik} used in the different re-rankers; space limitations prevent us from describing them in further detail.

6 Experimental Results and Analysis

Table 4 shows the data used to train each stage, drawn from the ACE training data and other sources. The training samples of the re-rankers are obtained by running the name tagger in cross-validation. 100 ACE 04 documents were held out for use as test data.

In the following we evaluate the contributions of re-rankers in name identification and classification separately.

Model	Identification		
	Precision	Recall	F-Measure
Baseline	93.2	93.4	93.3
+name structure	94.0	93.5	93.7
+relation	93.9	93.7	93.8
+event	94.1	93.8	93.9
+cross-doc coreference	95.1	93.9	94.5

Table 5. Name Identification

Model	Classification Accuracy	Identification +Classification		
		P	R	F
Baseline	93.8	87.4	87.6	87.5
+name structure	94.3	88.7	88.2	88.4
+relation	95.2	89.4	89.2	89.3
+event	95.7	90.1	89.8	89.9
+cross-doc coreference	96.5	91.8	90.6	91.2

Table 6. Name Classification

Tables 5 and 6 show the performance on identification, classification, and the combined task as we add each re-ranker to the system.

The gain is greater for classification (2.7%) than for identification (1.2%). Furthermore, we can see that the gain in identification is produced primarily by the name structure and coreference components. As we noted earlier, the name structure analysis can correct boundary errors by preferring names with complete internal components, while coreference can resolve a boundary ambiguity for one mention of a name if another mention is unambiguous. The greatest gains were therefore obtained in boundary errors: the stages together eliminated over 1/3 of boundary errors and about 10% of spurious names; only a few missing names were corrected, and some correct names were deleted.

Both relations and events contribute substantially to classification performance through their selectional constraints. The lesser contribution of events is related to their lower frequency. Only 11% of the sentences in the test data contain instances of the original ACE event types. To increase the impact of the event patterns, we broadened their coverage to include additional frequent event types, so that finally 35% of sentences contain event "trigger words".

We used a simple cross-document coreference method in which the test documents were clustered based on their cross-entropy and documents in the same cluster were treated as a single document for coreference. This produced small gains in both identification (0.6% vs. 0.4%) and classification (0.8% vs. 0.4%) over single-document coreference.

7 Discussion

The use of 'feedback' from subsequent stages of analysis has yielded substantial improvements in name tagging accuracy, from F=87.5 with the baseline HMM to F=91.2. This performance compares quite favorably with the performance of the human annotators who prepared the ACE

2005 training data. The annotator scores (when measured against a final key produced by review and adjudication of the two annotations) were F=92.5 for one annotator and F=92.7 for the other.

As in the case of the automatic tagger, human classification accuracy (97.2 - 97.6%) was better than identification accuracy (F = 95.0 - 95.2%).

In Figure 5 we summarize the error rates for the baseline system, the improved system without coreference based re-ranker, the final system with re-ranking, and a single annotator.[8]

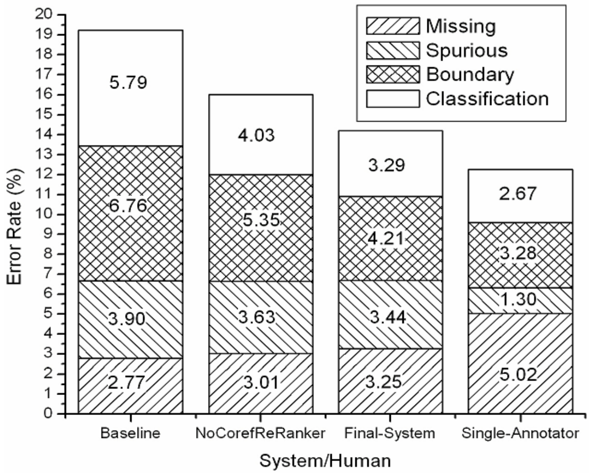

Figure 5. Error Distribution

Figure 5 shows that the performance improvement reflects a reduction in classification and boundary errors. Compared to the system, the human annotator’s identification accuracy was much more skewed (52.3% missing, 13.5% spurious), suggesting that a major source of identification error was not difference in judgement but rather names which were simply overlooked by one annotator and picked up by the other. This further suggests that through an extension of our joint inference approach we may soon be able to exceed the performance of a single manual annotator.

Our analysis of the types of errors, and the performance of our knowledge sources, gives some indication of how these further gains may be achieved. The selectional force of event extraction was limited by the frequency of event patterns – only about 1/3 of sentences had a pattern instance. Even with this limitation, we obtained a gain of 0.5% in name classification. Capturing a broader range of selectional patterns should yield further improvements. Nearly 70% of the spurious names remaining in the final output were in fact instances of 'other' types of names, such as book titles and building names; creating explicit models of such names should improve performance. Finally, our cross-document coreference is currently performed only within the (small) test corpus. Retrieving related articles from a large collection should increase the likelihood of finding a name instance with a disambiguating context.

Acknowledgment

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023, and the National Science Foundation under Grant IIS-00325657. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views 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. ANLP1997*. pp. 194-201., Washington, D.C.

Jianfeng Gao, Mu Li, Andi Wu and Chang-Ning Huang. 2005. Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach. *Computational Linguistics 31(4)*. pp. 531-574

Heng Ji and Ralph Grishman. 2005. Improving Name Tagging by Reference Resolution and Relation Detection. *Proc. ACL2005*. pp. 411-418. Ann Arbor, USA.

Heng Ji, Cynthia Rudin and Ralph Grishman. 2006. Re-Ranking Algorithms for Name Tagging. *Proc. HLT/NAACL 06 Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*. New York, NY, USA

Dan Roth and Wen-tau Yih. 2004. A Linear Programming Formulation for Global Inference in Natural Language Tasks. *Proc. CONLL2004.*

Dan Roth and Wen-tau Yih. 2002. Probabilistic Reasoning for Entity & Relation Recognition. *Proc. COLING2002.*

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)*

[8] Here *spurious* errors are names in the system response which do not overlap names in the key; *missing* errors are names in the key which do not overlap names in the system response; and *boundary* errors are names in the system response which *partially* overlap names in the key plus names in the key which partially overlap names in the system response.