An Adaptive Approach to Named Entity Extraction for Meeting Applications

Fei Huang, Alex Waibel
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh,PA 15213
fhuang@cs.cmu.edu, ahw@cs.cmu.edu

ABSTRACT

Named entity extraction has been intensively investigated in the past several years. Both statistical approaches and rule-based approaches have achieved satisfactory performance for regular written/spoken language. However when applied to highly informal or ungrammatical languages, e.g., meeting languages, because of the many mismatches in language genre, the performance of existing methods decreases significantly.

In this paper we propose an adaptive method of named entity extraction for meeting understanding. This method combines a statistical model trained from broadcast news data with a cache model built online for ambiguous words, computes their *global context* name class probability from *local context* name class probabilities, and integrates name lists information from meeting profiles. Such a fusion of supervised and unsupervised learning has shown improved performance of named entity extraction for meeting applications. When evaluated using manual meeting transcripts, the proposed method demonstrates a 26.07% improvement over the baseline model. Its performance is also comparable to that of the statistical model trained from a small annotated meeting corpus. We are currently applying the proposed method to automatic meeting transcripts.

Keywords

named entity extraction, meeting application, cache model

1. INTRODUCTION

Named entity extraction, i.e., extracting specific information structures, such as proper names, time and numerical expressions, from written or spoken languages, has been intensively investigated in the past several years [8, 6]. Much of the previous work on name finding is based on one of the following approaches: (1) handcrafted or automatically acquired rules or finite state transducers([1, 2]); (2) lookup from large name lists or other specialized resources [11]; (3) statistical models[3, 5]. Both statistical approaches and rule-based approaches have achieved very good performance, with 93 F-score on written English newswire articles compared with the 97 F-score achieved by humans[4], where F-score is a combined measure of precision and recall. When applied to manual broadcast news (**BN**) transcripts (0% WER), the F-score is only slightly lower, at 91[10]. However when applied to languages that are highly informal and contain strong spoken language characteristics, the performance of existing methods degrades significantly.

Meeting understanding is one application with these language characteristics. Unlike written newswire articles, meeting transcripts/hypotheses have no case and punctuation information available to facilitate named entity extraction. Even compared with broadcast news (**BN**), the language used in meetings(**MT**) is more informal[13]. Generally speaking, meeting language is characterized by:

- Shorter sentences,
  (7.7 words per sentence in MT vs. 12.1 in BN);
- Fewer sentences per turn,
  (2.2 in MT vs. 4.1 in BN);
- More disflunecies,
  (0.87 per sentence in MT vs. 0.48 in BN);
- More repairs,
  (29.0% in MT vs. 13.8% in BN);
- More non-lexicalized filled pauses, e.g., “uh,um”,
  (29.5% in MT vs.0.7% in BN).

Considering the many mismatches in language genre, the rules and statistical models acquired from formal written or spoken languages can not be directly used for meeting applications. Unfortunately it would be very expensive to record lots of meetings, transcribe them and build a corpus for training. Therefore we propose an adaptive method for named entity extraction for meeting understanding. This method is based on a statistical model trained from broadcast news data, but additionally makes use of meeting level global context information and name lists information in meeting

.

profiles to improve performance. This combination of supervised and unsupervised learning proved to be very effective. The experimental result on manual meeting transcripts is significantly better than the baseline model, which is trained from a large annotated broadcast news corpus. Its performance is also comparable to that of a statistical model trained from a small annotated meeting corpus.

In the following sections, we will introduce the baseline statistical model (Section 2), describe the proposed method (Section 3), present the experimental results and analysis (Section 4), and finally draw some conclusions (Section 5).

2. BASELINE MODEL

The baseline model adopts the Hidden Markov Model framework described in [3]. Eight internal states represent 7 classes of named entities (*PERSON,LOCATION, ORGANIZATION, TIME, DATE, MONEY* and *PERCENT*) as well as one remaining class (*NOT_A_NAME*). This generative model assumes the following generation process of a given sentence:

- current name class NC is selected according to the previous word and its name class,
- the first word in a name class is generated according to the current and previous name classes,
- each subsequent word in this name class is generated from a class-dependent bigram model.

Thus the task in the training procedure is to estimate 3 probabilities:

1. $P_c(NC|w_{-1}, NC_{-1})$, class transition probability,
2. $P_f(w_1|NC, NC_{-1})$, first word generation probability,
3. $P_b(w|w_{-1}, NC)$, class-dependent bigram probabilities.

In the above notations, NC and NC_{-1} represent the current and previous name classes respectively, w_1 represents the first word in the current name class, w represents the current word, and w_{-1} represents the previous word. To deal with data sparseness, different smoothing techniques, such as back-off and interpolation strategies, are used in the baseline model.

In the decoding process, the Viterbi decoding algorithm [12] is applied to find the name class sequence which maximizes the probability of generating the whole sentence with L words,

$$\begin{aligned}
\mathbf{NC} &= argmax_{\vec{NC}}\ P(\vec{W}, \vec{NC}) && (1)\\
&= argmax_{\vec{NC}}\ p(NC_1) \times p(w_1|NC_1) \times \\
&\quad \prod_{i=2}^{L} \tilde{p}(w_i, NC_i|w_{i-1}, NC_{i-1}), && (2)
\end{aligned}$$

where $\vec{W}$ stands for word sequence $(w_1, w_2, \ldots, w_L)$, $\vec{N}$ denotes name class sequence $(NC_1, NC_2, \ldots, NC_L)$, and $\tilde{p}(w_i, NC_i| w_{i-1}, NC_{i-1})$ represents the transition probability from w_{i-1} to w_i, assuming class transition from NC_{i-1} to NC_i.

When the transition is between different classes,

$$\begin{aligned}
\tilde{p}(w_i, NC_i|w_{i-1}, NC_{i-1}) &= P(end|w_{i-1}, NC_{i-1}) \times \\
P_c(NC_i|w_{i-1}, NC_{i-1}) &\times P_f(w_i|NC_i, NC_{i-1}). && (3)
\end{aligned}$$

When the transition is within the same class, i.e., $NC_i = NC_{i-1}$,

$$\begin{aligned}
\tilde{p}(w_i, NC_i|w_{i-1}, NC_{i-1}) &= \\
P(no\_end|w_{i-1}, NC_{i-1}) &\times P_b(w_i|w_{i-1}, NC_i). && (4)
\end{aligned}$$

The $P(end|w_{i-1}, NC_{i-1})$ and $P(no\_end|w_{i-1}, NC_{i-1})$ denote the probability of exiting or remaining in the previous name class given the previous word.

Working with *spoken* language, a lot of informative format information, like punctuation, case information and Arabic numerical expression (e.g.,10/23/1997), is no longer available. Therefore, our model only considers words as the basic modeling unit, and disregards format feature representations. This is different from BBN's IdentiFinder system[3].

As in most statistical NLP work, data sparseness is also a serious problem. We adopt both interpolation strategies (interpolate the best-fit model with the more general model to reduce over-fitting), and back-off strategies for smoothing. Back-off paths for each of the three probabilities are:

- $P_c(NC|w_{-1}, NC_{-1}) \rightarrow P_c(NC|NC_{-1}) \rightarrow P_c(NC) \rightarrow \frac{1}{number\_of\_name\_classes}$
- $P_f(w_1|NC, NC_{-1}) \rightarrow P_f(w_1|NC) \rightarrow P_f(w_1) \rightarrow \frac{P(NC)}{Vocabulary\_size}$
- $P_b(w|w_{-1}, NC) \rightarrow P_b(w|NC) \rightarrow \frac{P(NC)}{Vocabulary\_size}$

When tested on broadcast news data, the performance of the baseline model is comparable to the IdentiFinder. Detailed results are presented in Section 4.

3. ADAPTATION MODEL

To deal with the mismatch between formal and informal language, the proposed adaptation model uses the meeting level global context information as well as meeting profile information to improve the performance of named entity extraction.

3.1 Unsupervised Adaptation: Cache model

Cache models were first proposed for dynamic language modeling for speech recognition in [7], where the pre-computed general trigram model is interpolated with the local trigram model. The so-called *cache model* is trained from on-line generated word hypothesis, to “capture the short-term fluctuation in word frequency”[7]. Our cache model adaptation procedure also makes use of global context information from whole meeting transcripts, e.g., topical information, consistency of name class for a given word, to improve name class annotation in local contexts, but with different theory and model implementations.

The basic assumption of the proposed model is that each meeting will have some coherent topics, and even if a word or word sequence could have more than one name class in general, the name class of its every instance in a specific context (e.g. throughout a meeting) will tend to be consistent. This class will be in line with the topic of the scenario. Although the topic mismatches and the disfluency of spoken language in meetings will reduce the accuracy of the probability estimation, the average probabilities over the whole meeting, which is supposed to be internally coherent, may help give a reliable estimation, and possibly correct some errors in annotation. Therefore, the adaptation model will identify ambiguous words from first-pass annotation, build a cache to store their probability of belonging to each name class in each instance, estimate their global name class probability, and then relabel their name classes accordingly.

Formally, for a given word w, the best name class estimation in terms of the whole meeting context should satisfy

$$\begin{aligned} N\hat{C}(w) &= argmax_{NC}\ P(NC|w) \\ &= argmax_{NC}\ \prod_i P(NC_i = NC|w_i = w). \end{aligned} \quad (5)$$

$P(NC|w)$, the *global name class probability* for word w, is computed from the product of its *local name class probability* at position i, $P(NC_i = NC|w_i = w)$, under the independent assumption. The latter could be represented as the composition of 2 probabilities: *forward* NC probability $P(NC_i|w_i, w_{i-1})$ and *backward* NC probability
$P(NC_i|w_i, w_{i+1})$. This model tries to estimate the current class probability from its past and future context.

For forward probability,

$$P(NC_i|w_i, w_{i-1}) = \frac{P(w_i, NC_i|w_{i-1})}{P(w_i|w_{i-1})}, \quad (6)$$

where

$$\begin{aligned} P(w_i, NC_i|w_{i-1}) &= \frac{P(w_i, NC_i, w_{i-1})}{P(w_{i-1})} \\ &= \frac{\sum_{NC'_{i-1}} P(w_i, NC_i, w_{i-1}, NC'_{i-1})}{P(w_{i-1})} \\ &= \frac{\sum_{NC'_{i-1}} P(w_i, NC_i|w_{i-1}, NC'_{i-1})P(w_{i-1}, NC'_{i-1})}{P(w_{i-1})} \\ &= \sum_{NC'_{i-1}} \tilde{p}(w_i, NC_i|w_{i-1}, NC'_{i-1})p'(NC'_{i-1}|w_{i-1}) \end{aligned} \quad (7)$$

and

$$P(w_i|w_{i-1}) = \sum_{NC'_i} P(w_i, NC'_i|w_{i-1}). \quad (8)$$

For backward probability,

$$\begin{aligned} P(NC_i|w_i, w_{i+1}) &= \frac{P(w_{i+1}, w_i, NC_i)}{P(w_{i+1}, w_i)} \\ &= \frac{P(w_{i+1}|w_i, NC_i)P(w_i, NC_i)}{P(w_{i+1}, w_i)} \\ &= \frac{[\sum_{NC'_{i+1}} \tilde{p}(w_{i+1}, NC'_{i+1}|w_i, NC_i)]p'(NC_i|w_i)}{P(w_{i+1}|w_i)} \end{aligned} \quad (9)$$

In the above functions, $\tilde{p}$ is the transition probability, and $p'(NC|w)$ is a context-independent *prior* name class probability for word w, which is computed from the general domain broadcast news training data.

Thus, the local name class probability for word w at position i is the interpolation between the forward and backward probabilities, and w's name class probability over the whole meeting is the average probability over all the occurrences of word w. Such a global probability will be utilized for the re-estimation of w's name classes.

In summary, the whole name class annotation proceeds in the following way:

- Apply the baseline model on the test data;
- Identify ambiguous words, which have both:
  - different class labels over the whole meeting according to the first-pass decoding;
  - low class assignment confidence, which is defined in terms of the ratio between top 2 class-dependent word generation probabilities;
- Apply cache model re-estimation on those ambiguous words to compute their global name class probability;
- Select the winning class, which has the highest global name class probability, weighted by the frequency of that class label in the first-pass decoding;
- Relabel the ambiguous words with the winning class label.

3.2 Supervised Adaptation: Learning from Meeting Profile

Cache model adaptation works well when the true name class has the highest average probability among all labeled name classes after the first-pass decoding. However, when indicative information is not available (particularly for some OOV words), and thus first-pass labels are mostly incorrect, the model becomes less effective. However, some indicative information could be extracted from meeting profiles, which usually contain the attendants' names, the topics to be discussed, or even a concise summary of the meeting. When such information is taken into the model in the form of probabilistic name lists (e.g., person/location/organization name lists), in which each word is associated with the *prior* probability of belonging to this name class, more certainty about class annotation is obtained and therefore the named entity extraction performance

Table 1: Baseline model on BN and MT data

	BN	MT1	MT2
IdF	87.91	27.14	47.03
Baseline	88.35	37.93	60.37

Table 2: Adaptation on baseline model for MT data I

	MT1	MT2
BL	37.93	60.37
BL+MP	50.07	65.65
BL+MP+CM	66.67	68.33

will accordingly be improved.

In our current implementation, only attendees' name information is added to the meeting profile, and assigned to the *PERSON* name class with probability 0.9. The remaining probability mass, 0.1, is equally distributed among the rest name classes. These class-dependent unigram probabilities, $P(w|NC)$, are used in the computation of word generation probabilities.

4. EXPERIMENTAL RESULT

To evaluate the performance of the baseline model and the adaptation approach, we performed several experiments. We trained our baseline model using Hub4 NE-IE training data (52 broadcast news transcripts, about 260K words), and tested it on one manual broadcast news transcripts (2318 words, 106 named entities), obtained the **Baseline** result. We also ran the IdentiFinder (re-trained with the same broadcast news training data)on the same test data, obtained **IdF** result. Then, we ran the same experiment on two manual meeting transcripts, **MT1** (10554 words,137 named entities) and **MT2** (11146 words, 240 named entities). Table 1 summarizes the F-scores of these experiments.

As shown in Table 1, both IdentiFinder and the baseline model work reasonably well on broadcast news data, but their performances drop considerably when tested on the meeting data. Furthermore, their performances vary from meeting to meeting. This is understandable due to the various mismatches in language characteristics and the nature of different meetings.

Table 3: Adaptation on baseline model for MT data II

	BL	BL+MP+CM	Improvement	IdF(retrained)
MT1	37.93	66.67	75.77%	67.90
MT2	60.37	68.33	13.18%	61.11
MT3	47.76	54.99	15.13%	56.99
MT4	53.61	59.49	10.96%	63.87
MT5	53.87	58.23	8.09%	69.69
MT6	38.98	52.18	33.86%	66.10
MT7	60.33	61.13	1.32%	58.27
MT8	27.57	58.60	112.55%	68.32
Avg.	**47.55**	**59.95**	**26.07%**	**64.03**

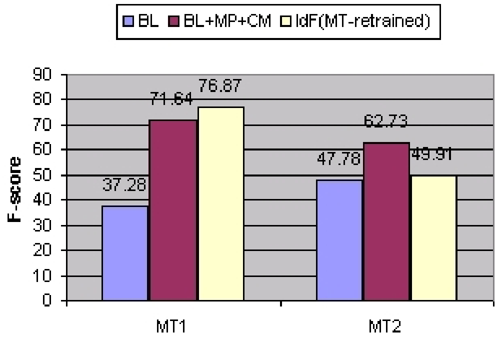

Figure 1: F-score comparison on ENAMEX class.

Table 2 demonstrates the experimental result when different adaptation strategies are applied. **BL**, **MP** and **CM** represent the baseline model, the meeting profile model and the cache model respectively. When the meeting profile information-in this case, the participant name lists-is integrated into the baseline model, the performance is improved, especially for person names. Specifically, in **MT1** the name list contains 45 instances of the 137 named entity instances, improving F-score by 32%, while in **MT2**, the name list contains 24 of the 240 named entity instances, improving F-score by 8.7%, respectively. Such difference in name list coverage also explains why name lists lead to more significant improvement in **MT1** than in **MT2**. When cache model adaptation is additionally applied on **BL+MP**, most of the local annotation errors are corrected as long as the true name classes are assigned higher probabilities on average during their baseline annotations. Thus performance is further improved. Figure 1 illustrates the F-scores of different systems on *ENAMEX*, which contains three name classes: *LOCATION*, *ORGANIZATION* and *PERSON*.

More experimental results are presented in Table 3, which shows that the cache model plus meeting profile information is very effective in **MT1**, **MT6** and **MT8**, and less effective in **MT7**. But in general, empirical experiments indicate that the proposed adaptive approach increases the named entity extraction accuracy by an average of 26.07% over the baseline system.

In Table 3, the performance of the proposed model is also compared with the IdentiFinder system (denoted as **IdF(retrained)**) retrained using a small number of manual meeting transcripts. Among all 8 meeting transcripts, which share similar topics and genres, 6 are left as training data, and the rest 2 are evaluated as test data. In each "fold" of such 4-fold cross validation experiments, the training set contains roughly same number (about 90K) of words, and includes most of the attendees' names in the test meetings, ranging from 58% to 100% instance coverage. Trained with such adaptation data, the **IdF** model demonstrates much better accuracy than the baseline system. Experimental result also shows that, although in general the performance of the proposed method, without using any adaptation data, is not as good as that of the IdF system trained with adaptation corpus, in some applications it is possible that the former is comparable(as in **MT1**, **MT3**, **MT4**), even outperforms

Example 1

BL: uh <b_enamex TYPE="PERSON">john <e_enamex> is on <b_enamex TYPE= "ORGANIZATION"> **channel one** <e_enamex> uh ***ty*** number two ***susi*** three ***nils*** four and myself five

BL+MP: uh <b_enamex TYPE="PERSON"> john <e_enamex> is on <b_enamex TYPE= "ORGANIZATION"> **channel one** <e_enamex> uh <b_enamex TYPE="PERSON"> ***ty*** <e_enamex> number two ***susi*** three ***nils*** four and myself five

BL+MP+CM: uh <b_enamex TYPE="PERSON"> john <e_enamex> is on **CHANNEL ONE** uh <b_enamex TYPE="PERSON"> ***TY*** <e_enamex> number two <b_enamex TYPE="PERSON"> ***SUSI*** <e_enamex> three <b_enamex TYPE="PERSON"> ***NILS*** <e_enamex> four and myself five

Reference: uh <b_enamex TYPE="PERSON"> john <e_enamex> is on channel one uh <b_enamex TYPE="PERSON"> ty <e_enamex> number two <b_enamex TYPE="PERSON"> susi <e_enamex> three <b_enamex TYPE="PERSON"> nils <e_enamex> four and myself five

Example 2

BL/BL+MP: is *bbn* a name
<b_enamex TYPE="PERSON"> ***bbn*** <e_enamex> 'S the name of a company
yeah then it 'S just *bbn* without spaces

BL+MP+CM: is *bbn* a name
BBN 'S the name of a company
yeah then it 'S just *bbn* without spaces

Reference: is <b_enamex TYPE="ORGANIZATION"> bbn <e_namex> a name
<b_enamex TYPE="ORGANIZATION ">bbn <e_enamex> 'S the name of a company
yeah then it 'S just <b_enamex TYPE="ORGANIZATION "> bbn <e_enamex> without spaces

Figure 2: Some examples excerpted from test data .

the latter(as in **MT2**, **MT7**).

Some segments excerpted from the test data are presented in Figure 2, where we can find that

- It is possible for the baseline model to detect common named entities, like "*john*", from informal/ungrammatical context, but the result resembles some named entity patterns from broadcast news, e.g., "*channel one*";
- Additional information from meeting profiles, although quite limited in amount, can be very helpful because of its high relevance to the meeting;
- Within the probabilistic model, name lists alone can not guarantee that every instance of their entries is correctly labeled, especially in the context with strong spoken language features, e.g."*...susi three nils four...*". But cache models can recover from such local annotation errors.
- The cache model adaptation works best when correct name classes are assigned higher probabilities on average. Otherwise — especially for the OOV words— it isn't helpful, even detrimental to the annotation, as in the case of "*bbn*".

5. DISCUSSION

5.1 Analogy to the forward/backward variables

The reader may have noticed the analogy between forward/backward probabilities and the $\alpha_t(i)$/$\beta_t(i)$ in forward-backward algorithm[9]. The forward variable$\alpha_t(i)$, the probability of generating the partial observation sequence $o_1, o_2, \ldots, o_t$ assuming the underlying state is i at time t, given the model λ, is defined as:

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i|\lambda). \quad (10)$$

The inductive computation for $\alpha_t(i)$ is

$$\alpha_t(j) \quad = \quad [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)], \quad (11)$$

where N is the number of states, a_{ij} is the transition probability from state i to state j, and $b_j(o_t)$ is the probability of generating output o_t at state j.

Similarly, the backward variable$\beta_t(i)$, the probability of generating the partial observation sequence $o_{t+1}, o_{t+2}, \ldots, o_T$, given the underlying state is i at time t,and the model λ. is defined as :

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T|q_t = i.\lambda), \quad (12)$$

The inductive computation for $\beta_t(i)$ is

$$\beta_t(i) \quad = \quad [\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)]. \quad (13)$$

Compared with formula (7), $\tilde{p}(w_i, NC_i|w_{i-1}, NC'_{i-1})$ and $a_{ij}b_j(o_t)$ both represent the probability of transiting between 2 states and generating new output in the new state, while $\alpha_{t-1}(i)$ and $p'(NC'_{i-1}|w_{i-1})$ denote the recursive probability corresponding to the previous state. Similarly, the analogy between backward probability and $\beta_t(i)$ also exists, but $\beta_{t+1}(j)$ is replaced by the constant prior probability $p'(NC|w)$.

5.2 Information retrieval based on meeting profiles

Meeting profiles usually contain limited information, and thus offer limited direct benefits. Nevertheless, some topic-related information in meeting profiles, e.g., scheduled topics, meeting summary, could be used as queries to retrieve relevant documents, on which the baseline model could be applied to extract topic-related named entities. Since most of the retrieved documents are written text, we would expect the baseline model attain more accurate annotations on them. Such additionally extracted named entities, together with their annotation confidence, could be integrated into the probabilistic name lists. This experiment will be carried out in the future.

5.3 Advantages and disadvantages

While the IdentiFinder system achieves good performance for general domain applications when enough training data is available, the proposed adaptive strategy works well with domain-dependent applications when no training data is available. This is achieved by the following:

- build a domain-independent model with general domain text;
- develop an adaptation model by "training on the fly", i.e., conducting supervised and unsupervised learning from test data;
- integrate the domain-independent model and the adaptation model for better named entity annotations.

However, utilizing this model prerequisites higher average probabilities for correct name classes, and assumes each word has one name class throughout the meeting, which may not be the case in some situation.

6. CONCLUSIONS

In this paper, we proposed an adaptive method of named entity extraction. This method combines a statistical domain-independent model with an adaptation model trained on the fly. For the latter, we build a cache model for ambiguous words, re-estimate their global context name class probability from local context name class probabilities, and integrate additional information from meeting profiles. We have demonstrated that such a combination of supervised and unsupervised learning greatly increases the performance of named entity extraction for manual meeting transcripts.

7. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, thank Jie Yang and other members in Interactive System Labs for their inspiring discussions and support, especially thank the data collection group for collecting, transcribing and annotating the meeting data. This research is supported by the DARPA TIDES project.

8. REFERENCES

[1] D. Appelt, J. Hobbs, D. Israel, and M. Tyson. Fastus: A finite-state processor for information extraction from real world texts. In *Proceeding of IJCAI-93*, 1993.

[2] S. Baluja, V. O. Mittal, and R. Sukthankar. Applying machine learning for high performance named-entity extraction. In *Pacific Association for Computational Linguistics*, 1999.

[3] D. Bikel, S. Miller, R. Schwarz, and R. Weischedel. Nymble: A high-performance learning name-finder. In *Proceedings of Applied Natural Language Processing*, pages 194–201, 1997.

[4] N. Chinchor. Overview of muc-7/met-2. In *Proceedings of the Seventh Message Understanding Conference(MUC7), http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html*, 1998.

[5] Y. Gotoh and S. Renals. Information extraction from broadcast news. In *Philosophical Transactions of the Royal Society of London*, A 358, pages 1295–1310, 2000.

[6] R. Grishman and B. Sundheim. Design of the muc-6 evaluation. In *Proceedings of MUC-6*, 1995.

[7] R. Kuhn and R. D. Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(6):570–583, 1990.

[8] N.Chinchor, P. Robinson, and E. Brown. The hub-4 named entity task definition, version 4.8. In *Proceedings of DARPA Broadcast News Workshop http://www.nist.gov/speech/hub4_98*, 1999.

[9] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Feburary,1989.

[10] P. Robinson, E. Brown, J. Burger, N. Chinchor, A. Douthat, L. Ferro, and L. Hirschman. Overview: Information extraction from broadcast news. In *Proceedings of DARPA Broadcast News Workshop*, pages 27–30, 1999.

[11] M. Stevenson and R. Gaizauskas. Using corpus-driven name lists for name entity recognition. In *Proceedings of 6th Applied Natural Language Processing and 1st North American Chapter of the Association for Computational Linguistics*, 2000.

[12] A. Viterbi. Error bound for convolutional codes and asymptotically optimum decoding algorithm. *IEEE Transaction on Information Theory*, 13:260–269, 1967.

[13] K. Zechner. Automatic summarization of spoken dialogs in unrestricted domains. In *Ph.D Thesis, Language Technology Institute, Carnegie Mellon University*, 2001.