Bootstrapping Named Entity Recognition for Italian Broadcast News

Marcello Federico Nicola Bertoldi Vanessa Sandrini
ITC-irst - Centro per la Ricerca Scientifica e Tecnologica
38050 Povo, Trento - Italy
{federico,bertoldi,sandrini}@itc.it

Abstract

This paper presents the development of a Named Entity (NE) recognition system for the Italian broadcast news domain. A statistical model is introduced based on a trigram language model defined on words and NE classes. The estimation of the NE model is carried out with a very little list of 2,360 manually tagged NEs and a large untagged newspaper corpus. An iterative training procedure is applied which goes through the estimation of simpler models, whose parameters are used to initialize the complete NE model. In the end, NE recognition experiments are reported, on broadcast news transcripts generated by a speech recognition system.

1 Introduction

Named Entity (NE) recognition is the task of spotting and classifying proper names and numerical expressions inside written or spoken documents. Research on spoken NE recognition has been mainly carried out on American English Broadcast News (BN), within the framework of DARPA-sponsored evaluations[1], since 1998. In the standard way, NE recognition in BN is performed on transcripts generated by a speech recognition system. In contrast to written documents, automatic transcripts may contain recognition errors and disfluencies, typical of the spoken language, and lack punctuation and capitalization. The presence of noise and the absence of important textual clues in the source document seem good reasons for approaching spoken NE recognition with statistical or data-driven methods.

This paper addresses the issue of developing, from scratch, a NE recognition system for the Italian BN domain. In particular, no tagged training corpus will be used to accomplish this task.

A trigram NE Language Model (LM) of words and NE classes is bootstrapped by just exploiting two relatively cheap written-language resources: a 240M-word corpus of newspaper articles and a list of 2,360 NEs, which were manually classified into possible NE classes.

In the NE LM, each class is modelled as a combination of two probabilistic finite state LMs: a template model and a bag-of-word model of known entries of the class, both of which can span more words. Training of the NE LM goes through the estimation of simpler models, which are then used to initialize the full model. Finally, the NE LM is adapted to work with automatic transcripts by inhibiting the template model, and by removing capitalization and punctuation information from the model.

Section 2 describes the structure and parameters of the NE LM, Sections 3 and 4 explain, respectively, how NE recognition and parameter estimation can be performed with the model. Section 5 shows experimental results. Finally, Sections 6 and 7, respectively, contain some discussion and conclusions about the here presented approach.

[1] http://www.nist.gov/speech/tests/

Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 296-303.
Association for Computational Linguistics.

Σ	set of words (either up-case or low-case) in the corpus
$\mathcal{V} \in \Sigma$	word vocabulary of the language model
$\mathcal{T}$	strings matched by the template
$\mathcal{E} = \{loc, per, org, oth\}$	set of named entity categories
$x, y, z \in \mathcal{V} \cup \mathcal{E}$	words or named entity categories
$e \in \mathcal{E}$	named entity category
$t \in \mathcal{T}$	any string matched by the template, i.e. potential named entity
$\Pi = \{\pi(x, y, z) : \ x, y, z \in \mathcal{V} \cup \mathcal{E}\}$	n-gram probabilities i.e. $\pi(x, y, z) = \Pr(z \mid xy)$
$\Lambda = \{\lambda(e) : \ e \in \mathcal{E}\}$	template vs. list probability, e.g. $\lambda(loc) = \Pr(< \text{tmpl} > \mid < loc >)$
$T = \{\tau_i : \ i = 1, \ldots, 4\}$	template probabilities
$E = \{\epsilon(t, e) : \ t \in \mathcal{T}, \ e \in \mathcal{E}\}$	string-class distributions $\epsilon(\texttt{Il Cairo}, < loc >) = \Pr(\texttt{Il Cairo} \mid loc)$

Table 1: List of often used symbols.

2 Statistical NE LM

The proposed NE LM generalizes an ordinary trigram LM as follows. Trigram probabilities are defined over a vocabulary $\mathcal{V}$, including common words, and a set of NE categories $\mathcal{E} = \{loc, per, org, oth\}$[2]. Moreover, a further class *oov* (out-of-vocabulary) is defined to capture words not belonging to $\mathcal{V}$. In Figure 1, for the sake of clarity, the Probabilistic Finite State Network (PFSN) corresponding to a bigram NE LM is shown. Notice that in all the here depicted networks, transitions or arches with probabilities corresponding to parameters to be estimated are drawn with solid lines, while transitions with fixed probabilities are drawn with dashed lines. Moreover, *initial* states or nodes are in white colour, *final* states are in black colour, and other states are in gray colour. Labels on transitions may either correspond to strings to be matched with the input text, or to names of other networks, e.g. `<loc>`. Transitions without label are called empty. As no recursion will be used in the subsequent PFSNs, the NE LM can be more properly defined as a cascade of PFSNs.

Each NE class has associated a binary random variable which switches between a bag-of-word distribution of known-entries for that class, and a distribution of generic NE templates. The PFSNs corresponding to the switch variable and to the known-entry distribution are shown, for class *loc*, in Figure 2.

Finally, generic NE templates are modelled with

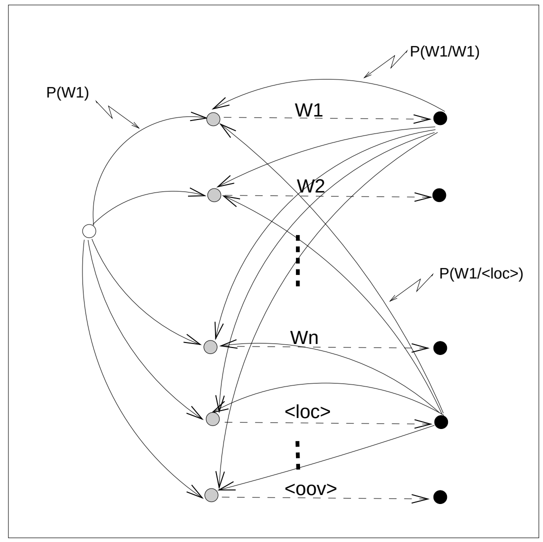

Figure 1: Network of a bigram NE LM.

regular expression covering almost all possible ways of writing proper names. Let `<Word>` denote any up-case word, and `<prep>` any preposition that may occur before or between proper names [3]. Four sub-cases of the generic regular expression:

$$(< \texttt{prep} >? < \texttt{Word} >)+$$

have been identified (see Figure 3), which correspond to different lengths and formats of NEs. In the resulting PFSN, see Figure 3, probabilities associated to each case are tied among all NE classes. For reason of brevity, PFSNs of

[2] NE recognition is here limited to proper nouns of type *location*, *person*, and *organization*, plus a filler class for other types of proper nouns (cf. Section 5.2).

[3] A list of 38 preposition was defined for this purpose.

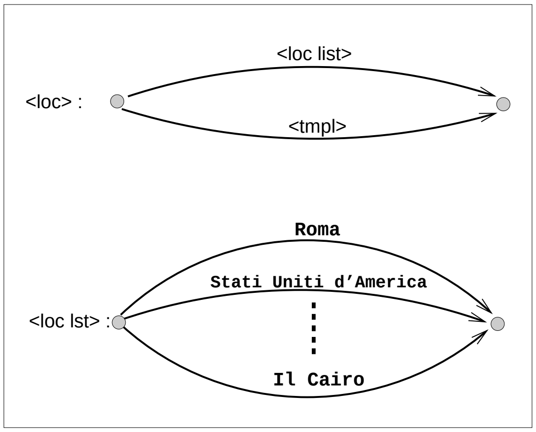

Figure 2: Switch variable and bag-of-word model for NE class *loc*.

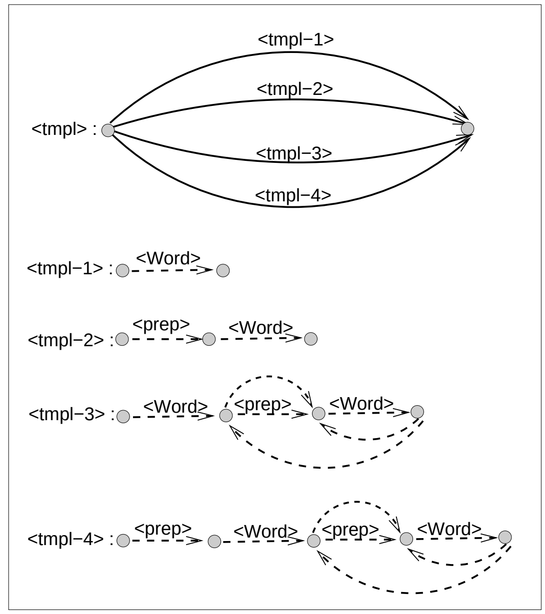

Figure 3: NE class: template model.

words in $\mathcal{V}$, and of labels `<oov>`, `<Word>`, and `<prep>` are omitted.
Hence, the described model can be summarize by a 4-tuple of parameters (Π, Λ, T, E), defined as follows (cf. definitions in Table 1):

- Π, the set of trigram probabilities $\pi(x, y, z) = \Pr(z \mid xy)$, $x, y, z \in \mathcal{V} \cup \mathcal{E}$;
- Λ, the template vs. list probability associated to each NE class, $\lambda(e) \in [0, 1]$, $e \in \mathcal{E}$;
- T, the probabilities associated to the templates, $\tau_i \in [0, 1] \quad i = 1, 2, 3, 4$;
- E, the probabilities of known-entries of each class $\epsilon(t, e) = \Pr(t \mid e)$, $t \in \mathcal{T}$, $e \in \mathcal{E}$.

While probabilities in Λ, T, E are defined by discrete probability schemes, those in Π are smoothed in order to cope with data sparseness. Hence, the following interpolation scheme coupled with an absolute discounting method (Huang et al., 2001) is used:

$$\begin{aligned}\Pr(z \mid xy) &= \max\left\{\frac{c(xyz) - 1}{c(xy)}, 0\right\} + \frac{d(xy)}{c(xy)} \times \\ &\times \left\{\max\left\{\frac{c(yz) - 1}{c(y)}, 0\right\} + \frac{d(y)}{c(y)} \times \right. \\ &\times \left. \frac{c(z)}{N}\right\} \qquad (1)\end{aligned}$$

which employs the following statistics computed on the training sample: $c(\cdot)$, the event counts; $d(\cdot)$, the number of different words observed after a specified bigram or unigram; N, the size of the training sample.

3 NE Recognition

Given a text W, its probability can be computed through the LM PFSN by taking into account all possible paths (sequence of states) that exactly span W, start from the initial state, and end in one final state. It is easy to see that if paths trace all the traversed sub-networks, they correspond to parse trees, which indicate, at the top level, eventually recognized NEs.
However, for the sake of NE recognition, one is more interested in finding the most probable parse tree T^* for W, i.e.:

$$T^* = \arg\max_{T \in T(W)} \Pr(W, T) \qquad (2)$$

where $T(W)$ indicates the set of paths described in the begin of this section. It can be shown that the here represented NE LM is equivalent to a cascade of hidden Markov models (HMMs)

(Huang et al., 2001) or of weighted finite state acceptors (Pereira et al., 1994) and that the most probable path for an input text can be efficiently computed through a Viterbi-like decoding algorithm (Brugnara and Federico, 1997). Here, in particular, the decoding software used for speech recognition is used by converting the input front-end to a stream of ASCII characters, and by replacing acoustic models to single state HMMs with delta distributions over single ASCII characters. The NE LM is compiled into a set of distinct PFSNs, corresponding to the main trigram LM, the class related models, etc. Significant memory savings are achieved by exploiting a tree-based topology for the trigram and bag-of-word models (Bertoldi et al., 2001; Huang et al., 2001).

4 NE LM Training

Given a manually parsed corpus, Maximum Likelihood (ML) estimation of the NE LM just requires collecting sufficient statistics for its parameter sets. Otherwise, if just an untagged text is available, training of the LM can be performed by the Expectation-Maximization algorithm (Dempster et al., 1977) or, more easily, by the Viterbi training method, also known as segmental K-means algorithm (Juang and Rabiner, 1990).

4.1 Viterbi training

Let $\tilde{M}$ be an available estimate of the NE LM and W an untagged text. A new estimate $\hat{M}$ can be obtained by searching for the best parse tree $\tilde{T}$, under $\tilde{M}$, and by computing, then, the ML estimate $\hat{M}$, under $\tilde{T}$. This corresponds to performing the two following steps:

1. $\tilde{T} = \arg\max_{T} \log \Pr(T, W; \tilde{M})$

2. $\hat{M} = \arg\max_{M} \log \Pr(\tilde{T}, W; M)$

The following inequalities show that the above procedure can be iteratively used to improve the likelihood of the best parse tree:

$$\begin{aligned} \max_{T} \log \Pr(T, W; \hat{M}) &\geq \log \Pr(\tilde{T}, W; \hat{M}) \\ &\geq \log \Pr(\tilde{T}, W; \tilde{M}) \\ &= \max_{T} \log \Pr(T, W; \tilde{M}) \end{aligned} \quad (3)$$

However, the above property does not tell if the parameter transformation $\tilde{M} \rightarrow \hat{M}$ indeed converges to a fixed point. A tricky convergence proof of the segmental K-means algorithm applied to HMMs can be found in (Juang and Rabiner, 1990), while bounds on the distance between HMM parameters estimated by EM and Viterbi training are discussed in (Merhav and Ephraim, 1991).
In this work, few iterations of Viterbi training were applied, as relative likelihood improvements of the best interpretation drastically reduced after the first iteration. Figure 4 shows how training is applied to the NE LM. Starting from some model estimate $\tilde{M}$, the corpus is tagged according to the most probable parse tree $\tilde{T}$. Hence, sufficient statistics are extracted from the tagged corpus in order to estimate $\hat{M}$. In some cases, little supervision in terms of manually checked NE lists is used to filter out unreliably tagged data.

4.2 Incremental training

Training of the NE LM goes through the definition and estimation of two simpler intermediate models, which are obtained by fixing some of the parameters in (Π, Λ, T, E). In particular, the notation $(\Pi, \Lambda = 0, E)$ indicates a *list model*, as the switch parameters in Λ inhibit the template part, while $(\Pi, \Lambda = 1, T)$ indicates a *template model*, as the lists of known-entries are inhibited [4]. Estimates of intermediate models will be used to initialize the complete NE LM.

4.3 Intermediate model M_0

A list model $(\Pi, \Lambda = 0, E)$ is estimated starting from a supervised list of NE entries which may also include ambiguous cases[5]. Initialization and estimation of distributions in Π and E are performed on a sub corpus made of sentences

[4]Not active parameters are just omitted from the notation.

[5]See Section 5.2 for details.

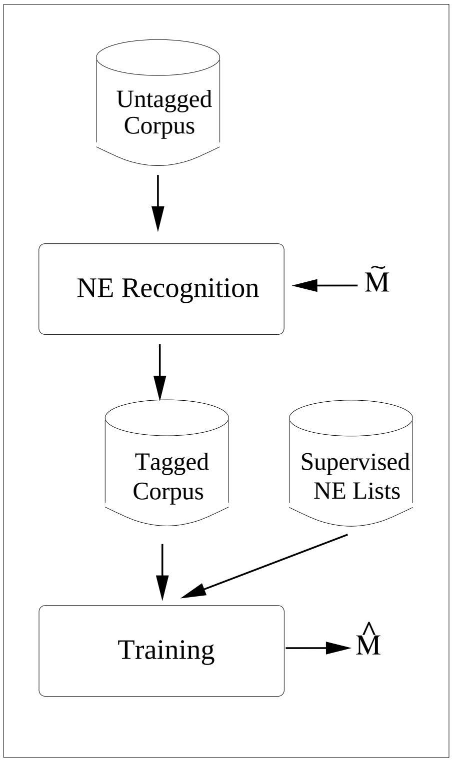

Figure 4: Viterbi training procedure.

not containing other potential NEs, but those in the supervised list. Then, Viterbi training is applied to annotate the whole corpus. This step permits to improve the initial estimates of Π and E.

4.4 Intermediate model M_1

The template model ($\Pi = \Pi_0, \Lambda = 1, T$) is estimated, with the probabilities in Π fixed with the final estimates of M_0, while T are uniformly initialized. This time, only a very small sample of the corpus is needed because model M_1 just requires estimating three free parameters.

4.5 Full model M_2

The complete model ($\Pi = \Pi_0, \Lambda, T = T_1, E$) is estimated, with E initialized with E_0, and with each $\lambda(e)$ initialized with the ratio of the number of new NEs, of type e, tagged by M_1, to the total number of NEs of type e tagged by M_1. This gives a rough estimate of the chance of finding NEs of a given class which are not in the list of known-entries. Training of this model permits to significantly expand the support sets of the distributions in E and to accordingly estimate the switch probabilities Λ for each class. Ideally, this means that new NEs can be discovered and stored by the distributions in E.

4.6 Spoken NE model M_3

As model M_2 exploits capitalization information in texts, for the template matching, it cannot be directly applied to case-insensitive automatic transcripts. Hence, the list model ($\Pi = \Pi_0, \Lambda = 0, E = E_2$) is derived from M_2, by inhibiting the template case and re-estimating Π on the tagged corpus used to estimate M_0, after removal of punctuation and capitalization information. Finally, capitalization is also eliminated from the entries in E_2.

5 Experimental results

5.1 Training and testing data

The training corpus is a 240M-word collection of Italian newspapers spanning the period 1992-1999, while the evaluation sets consists of two broadcast news shows, of November 1999, with a total duration of 40 minutes. To avoid time overlap with testing data, only written resource before October 1999 were considered for training purposes.

Reference transcripts of the test set were manually produced and include punctuation and capitalization for a total of about 7,000 words and 322 tagged NEs. NE recognition performance was computed with the scoring software of the 1999 IE-ER DARPA evaluation. In particular, the F-score was used which integrates agreement between reference and hypothesis according to *type*, *content*, *extension* of NEs.

Automatic transcripts were generated with the ITC-irst broadcast news transcription system in (Bertoldi et al., 2001), which features a beam-search Viterbi decoder, context dependent HMMs, and a 64K-word trigram LM. Two recognition passes were applied on the news shows, providing an average word error rate (WER) of 19.8%.

5.2 List of supervised NEs

About 2,700 potential NEs were extracted from the training data and manually classified according to the annotation guidelines in (Chinchor et al., 1999). In particular, the 2,673 most frequent strings, matching template `<tmpl-3>` (see Figure 3) in the newspaper corpus, were included. Moreover 30 proper names, which were found in BNs manual transcripts, were added to the list (e.g. television networks, news programs, etc.). Among the collected strings, 2,360 were manually classified into one or more of the following categories: *location*, *organization*, *person*, and *other*. The filler category *other* which includes proper names as events, products, etc. was not taken into account for the sake of evaluation.

Notice that no temporal and numeric expression have been considered here. These entities are in fact much less frequent than proper names (Przybocki et al., 1999) and would hence require a much larger test set for an experimental evaluation.

5.3 Model training

Before estimating M_0, the training corpus was deterministically tagged according to the list of supervised NEs. Next, single sentences in the corpus which contain potential NEs not covered by the supervised list were removed. After this step, the training corpus was reduced to 125M words. These data were used to estimate E, while only the subset of sentences containing univocally classified NEs were used to estimate Π. In particular, about 37M trigrams were used to train the NE LM, which was then compiled into a PFSN.

Next, one Viterbi iteration was carried out which permitted to reduce the average number of classes associated to NEs from 1.29 to 1.01, and to augment the size of the training data for parameters in Π by 10%. However, this training step resulted in a limited impact in performance: just 1% F-score relative improvement on the reference transcripts (see Table 3).

The following model M_1 permits to estimate probabilities of the template models. Interestingly, combination of trigrams and templates

Figure 5: Correlation of NE frequencies across eight years.

provided a much better starting point, i.e. 77.76 F-score (see Table 3). After one Viterbi training iteration, the F-score improved by 2.8%. Given that the same templates are shared among all the NE classes, the M_1 model gives an indirect measure of the tagging accuracy by the trigram LM component.

Parameter estimates resulting from models M_0 and M_1 are finally combined to initialize model M_2. In other words, lists of known NEs and templates can be exploited jointly. This initialization achieves an F-score of 85.92, which corresponds to relative improvements of 7.5% and 19.7%, respectively, over the template model (M_1) and the list model (M_0). Training of model M_2 was carried out trough a single Viterbi iteration using just the newspaper texts from January to October 1999. The rationale for this is that relevant new NEs occur more likely around the period of the test set. This is some way confirmed by Figure 5, which shows the correlation coefficient between relative frequencies of potential NEs found in May 1999 against those found in 1999, 1998, ... ,1992.

After training, performance of M_2 improved by only 1.3%, but coverage by the NE lists significantly increased (see Table 2).

5.4 Spoken NE Recognition

In Table 4, NE recognition experiments on automatic transcripts are reported for model M_0,

	E_0		E_2	
	# wrd	cov.	# wrd	cov.
loc	662	75.42	25,752	96.61
org	608	62.66	19,605	84.00
per	1,132	44.19	48,109	84.50
all	2,122	59.94	70,261	89.75

Table 2: Size of the NE lists of model M_0 and M_2 and coverage statistics with respect to test data.

	Initial	Final
M_0	71.10	71.78
M_1	77.76	79.94
M_2	85.92	87.17

Table 3: F-scores of models M_0-M_2 on the test set, before and after training.

after removing punctuation and capitalization information, and for model M_3. Three different settings of the BN transcription system were considered in order to evaluate NE recognition with different word error rates (namely `rec1`, `rec2`, and `rec3`). Moreover, as a reference, manual transcripts without punctuation and capitalization (`txt-i`) were also used.

Results on the reference transcripts (`txt-i`) show that the lack of punctuation and capitalization causes a 8.6% relative loss in performance, i.e. F-score drops from 87.17 to 81.12. This is mainly due to the increase in ambiguity caused by common words which may also occur in proper names. However, the incremental training procedure allowed for a significantly improvement over the initial model M_0, i.e. a 20% F-score relative improvement, from 67.42 to 81.12.

Experiments on automatic transcripts with different WERs show relative decreases in performance, with respect to `txt-i`, ranging between 10.4% and 13.0%, for WERs between 19.8% and 23.0%. The relative improvement between the initial and final models, M_0 and M_3, is around 15-16% for all automatic transcripts. The reason for the lower performance improvement may be that M_3 basically augments M_0 with less frequent proper names which are probably more difficult to recognize, given the statistical nature of the speech recognizer.

	txt-i	rec1	rec2	rec3
M_0	67.42	62.95	61.96	61.34
M_3	81.12	72.69	72.14	70.57
wer%	0.0	19.8	21.3	23.0

Table 4: F-score by models M_3 and M_0 on BN transcripts with different WERs.

6 Discussion

This section compares the here proposed NE LM with the *NE tagged LM*, presented in (Gotoh et al., 1999; Renals et al., 1999). The NE tagged LM uses a different decomposition of the probability $\Pr(W,T)$, which can be related to an ordinary class based trigram model, i.e.:

$$\Pr(W,T) = \prod_{i=1}^{n} \Pr(w_i, t_i \mid w_{i-2}t_{i-2}w_{i-1}t_{i-1})$$

where T now corresponds to a word-by-word tagging of W with classes in $\mathcal{E} \cup \{e_0\}$, with e_0 denoting the not-NE class. NE recognition with this model can also be performed by Viterbi decoding. However, this requires estimating probabilities in the space $(V \times (\mathcal{E} \cup \{e_0\}))^3$, in contrast to the probability space $(V \cup \mathcal{E})^3$ used by the NE LM.

Moreover, the cascade structure of the NE LM can span longer dependencies, i.e. across words and NE classes, than the NE tagged LM can do. On the other side, the latter model is probably more flexible in the composition of NEs. In other words, new NEs can be recognized by concatenating known entries. The capability of finding new NEs is, for what concerns the NE LM, limited to the template model. Hence, a possible improvement could be to replace the current bag-of-word model in M_3 with an n-gram model, estimated on the entries currently used in E.

Interestingly, similar levels of performance are reported in (Renals et al., 1999) when NE recognition is carried out on clean, case-insensitive texts and automatic transcripts (with 21%

WER), i.e. F-scores are 85.0 and 75.0, respectively. Of course, a true comparison between the two approaches should be perfomed on the same language and data.

7 Conclusion

This paper presented a statistical language model for NE recognition which was developed for the Italian broadcast news domain. The model integrates trigram statistics on words and NE classes, with probabilistic finite state language models. A bootstrap training technique is presented which permits to estimate the model by means of very inexpensive language resources: a large newspaper corpus and a few thousand manually classified NEs.

Experimental results were provided for two automatically transcribed broadcast news shows. Presented results are comparable with those obtained, on similar conditions, by NE taggers for American English broadcast news, which were trained on much more supervised data.

8 Acknowledgements

This work was carried out under the projects CORETEX (IST-1999-11876) and WebFAQ partially funded, respectively, by the European Commission and the *Fondo Unico della Provincia di Trento.*

References

N. Bertoldi, F. Brugnara, M. Cettolo, M. Federico, and D. Giuliani. 2001. From broadcast news to spontaneous dialogue transcription: portability issues. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, UT.

F. Brugnara and M. Federico. 1997. Dynamic language models for interactive speech applications. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, pages 2751–2754, Rhodes, Greece.

N. Chinchor, E. Brown, L. Ferro, and P. Robinson. 1999. 1999 Named entity Recognition Task definition. Technical Report Version 1.4, MITRE, Corp., August. http://www.nist.gov/speech/tests/ie–er/er_99/doc/ne99_taskdef_v1_4.ps.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38.

Y. Gotoh, S. Renals, and G. Williams. 1999. Named entity tagged language models. In *Proceedings of ICASSP*, volume I, pages 513–516, Phoenix, March.

X. Huang, A. Acero, H.-W. Hon, and R. Reddy. 2001. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall.

B.-H. Juang and L. R. Rabiner. 1990. The segmental k-means algorithm for estimating parameters of hidden Markov models. *IEEE Trans. Acoust., Speech and Signal Proc.*, ASSP-38(9):1639–1641.

N. Merhav and Y. Ephraim. 1991. Hidden Markov modeling using a dominant state sequence with application to speech recognitionof states. *Computer Speech and Language*, 5:327–339.

F. Pereira, M. Riley, and R. Sproat. 1994. Weighted rational transductions and their application to human language processing. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 262–267, Plainsboro, NJ.

M. A. Przybocki, J. G. Fiscus, J. S. Garofolo, and D. S. Pallet. 1999. 1998 HUB-4 Information Extraction Evaluation. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, VA.

S. Renals, Y. Gotoh, R. Gaizauskas, and M. Stevenson. 1999. Baseline IE-NE experiments using the Sprach/Lassie system. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, VA, February.