Named Entity Extraction from Word Lattices

James Horlock, Simon King

Centre for Speech Technology Research
University of Edinburgh, UK
J.Horlock@ed.ac.uk, Simon.King@ed.ac.uk

Abstract

We present a method for named entity extraction from word lattices produced by a speech recogniser. Previous work by others on named entity extraction from speech has used either a manual transcript or 1-best recogniser output. We describe how a single Viterbi search can recover both the named entity sequence and the corresponding word sequence from a word lattice, and further that it is possible to trade off an increase in word error rate for improved named entity extraction.

1. Introduction

Our motivation for using word lattices rather than 1-best recogniser output is that the goal of the recogniser in producing the 1-best transcription (low word error rate, with all word errors being counted equally) does not coincide with the goal of named entity extraction, where we are more interested in getting the entity words correctly recognised (and correctly tagged as entities) than getting the non-entity words correct.

The standard method of scoring named entity recognition is the F-measure: $F = 2PR/(P+R)$ where precision, P, is the ratio of correctly tagged entities to the total number of entities found and recall, R, is the ratio of correctly tagged entities to the total number of manually annotated entities. We use standard software for computing F-measures and quote them as percentages.

It has been said, e.g. in [2, 5], that F-measure is inversely proportional to the word error rate (WER) of a given speech transcript: the more accurate the transcript (lower WER), the higher the F measure. In this paper we show that, although this is broadly true, it is possible to trade off more word errors for a higher F measure.

1.1. Data

Whenever we refer to word lattices we mean lattices produced by the HTK system which uses cross-word triphone HMMs, MLLR-adaptation and a 4-gram language model. The audio data was automatically segmented prior to recognition: there is one word lattice per segment. These lattices were provided by Cambridge University. All results quoted are for the development set from the 1997 Hub4 NE task. All model parameters were estimated on the corresponding training set which does not intersect with the development set.

2. System description

We first introduce the model used and then show that this is a finite state machine, which forms the basis of our implementation.

The model we use is very similar to that developed by BBN [1] and Mitre [3]. However, our system is designed to work with word lattices rather than linear word strings (manual transcripts or 1-best recogniser output). In our notation, E_1^L refers to the entities $e_1, e_2, \ldots, e_L$ and W_1^L refers to the words $w_1, w_2, \ldots, w_L$. Entity e_l is the entity tag of word w_l.

The most apparent difference between a system for text and one for word lattices is that, since the word sequence is not given, instead of trying to maximise the conditional probability of the entities, given the words, $P(E_1^L|W_1^L)$, we maximise the joint probability of entities and words, $P(E_1^L, W_1^L)$. This is the same as maximising $P(E_1^L|W_1^L)P(W_1^L)$ – which means that working with word lattices will require not only a search to find the most likely E_1^L but also an additional search over all possible W_1^L, weighting by $P(W_1^L)$. The task is therefore to find the E_1^L and W_1^L which maximise $P(E_1^L, W_1^L)$:

$$\begin{aligned} P(E_1^L, W_1^L) &= P(W_1^L|E_1^L)P(E_1^L) \\ &= \prod_{i=1}^{L} P(w_i|W_1^L, E_1^L) \cdot \prod_{i=1}^{L} P(e_i|E_1^L) \end{aligned}$$

making standard N-gram independence assumptions:

$$P(E_1^L, W_1^L) = \prod_{i=1}^{L} P(w_i|W_{i-N+1}^{i-1}, E_1^L)P(e_i|E_{i-M+1}^{i-1})$$

assuming that $P(w_i|W_{i-N+1}^{i-1}, E_1^L)$ depends only on the current entity e_i and that N=3 (a trigram word sequence model) and M=2 (a bigram entity sequence model):

$$P(E_1^L, W_1^L) = \prod_{i=1}^{L} P(w_i|w_{i-2}, w_{i-1}, e_i)P(e_i|e_{i-1}) \quad (1)$$

2.1. Graphical representation

First, consider only finding the named entities in a given linear word string, rather than a lattice. Since all probabilities in equation 1 are conditioned on a limited amount of left context (they are N-gram probabilities), the product can be computed by a finite state machine (FSM) in which the probabilities of entities $P(e_i|e_{i-1})$, and the probabilities of words $P(w_i|w_{i-2}, w_{i-1}, e_i)$, are associated with the arcs and states respectively.

Figure 1 shows a FSM that can compute equation 1, where probabilities of entities are evaluated on the arcs, and probabilities of words are evaluated in the states. The arrows above the arcs show the direction of flow of the arcs, thus states are always entered on the left and exited on the right. There is a state for each entity type: Organisations, People, Places, Times,

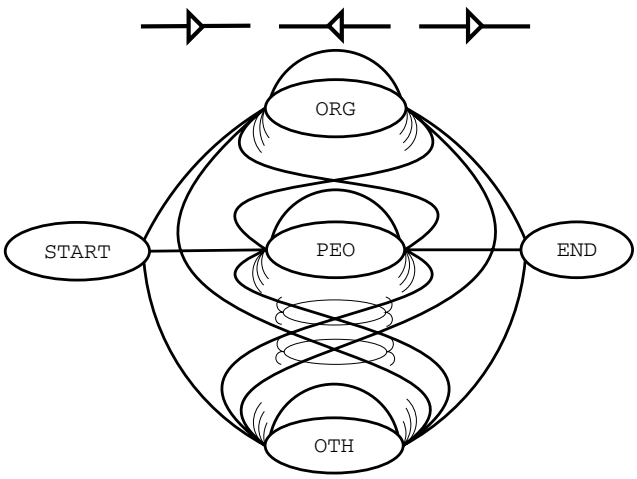

Figure 1: *Simplification of the topology used for the statistical model. Direction of transitions are indicated by arrows above the transitions to avoid confusion.*

Dates, Money, Percentages, plus one for the "not a named entity", Other.

To use this FSM to tag a word string, we must find a path (a state sequence) through the graph from START to END nodes, and in doing so, generate the required word sequence. This is a search problem and can be efficiently solved using Viterbi search. In the next section we describe how this process can be generalised to the case where the words to be generated are not a string but a lattice.

2.2. Extension to word lattices

The word lattice could be re-written as an N-best list of word strings, and each of the strings processed as above, but this would be inefficient since for typical word lattices a very long list of word strings would be produced. Instead, we note that both the word lattice and the FSM from above are finite state. In computing the sequences E_1^L and W_1^L which maximise equation 1, we must now traverse two finite state machines. To do this, we can construct a new finite state machine in which the states are pairs of states: one from the word lattice and one from the FSM above. Again, a Viterbi search can be used to find the most likely path through this graph – this path corresponds to the E_1^L and W_1^L which maximise equation 1.

2.2.1. Implementation

Rather than explicit construction of the FSM in which states are pairs of states from the two underlying FSMs, we implemented the search using the token passing algorithm [8] in which tokens consist of pairs of tokens, one in each of the FSMs. Application of the Viterbi criterion is as usual: when two or more tokens meet, and they are in the same *pair of states*, only the most likely one is retained. Search proceeds in synchronous fashion. In other words, all tokens are passed forward at the same time: each token in a pair takes a transition in its respective FSM.

3. Parameter estimation

The model requires the estimation of two different kinds of N-gram probabilities: word sequences and entity sequences.

3.1. Probabilities of entities

Estimating $P(e_i|e_{i-1})$, the probabilities on the arcs in figure 1, is easily achieved with a standard bigram language model, trained on manually annotated data.

3.1.1. Starting and ending entities

Our actual finite state machine is a little more complex than figure 1 because it is able to generate both

<PLACE>Edinburgh Scotland</PLACE>

and

<PLACE>Edinburgh</PLACE><PLACE>Scotland</PLACE>

for example, with different probabilities. However, this is not the focus of this paper, so will not describe this aspect of the system here.

3.2. Probabilities of words

The states of the model in figure 1 generate words. This is performed by backed-off trigram language models. Each model computes $P(w_i|w_{i-2}, w_{i-1}, e_i)$ with the conditioning on e_i being achieved by using entity-specific models - one in each state.

The language models were trained using the CMU-Cambridge language modelling toolkit [9], with Witten Bell discounting [6].

3.2.1. Starting and ending entities

Since the first and last words in an entity are very informative (i.e. they have different probability distributions to words in the middle of entities), this must be captured by the language models. We achieve this in the same way as standard language modelling for speech recognition, by introducing pseudo-words <s>and </s>. N-gram context is blocked by <s> – that is, if w_i is the first word in an entity, its probability is estimated as $P(w_i|\ <s>)$. If w_i is the last word in an entity, its probability is estimated as $P(w_i|w_{i-2}, w_{i-1})P(</s>\ |w_i)$.

3.2.2. Smoothing with recogniser language model probabilities

The lattices we are using were produced with an HMM-based recogniser with a 4-gram backed-off N-gram language model. These language model probabilities are available from the word lattice separately from the acoustic model probabilities. Therefore, we can use them to smooth our new estimates of $P(w_i|w_{i-2}, w_{i-1}, e_i)$ by linear interpolation in the log domain, before combination with the acoustic probability.

4. Other studies

The lattices available to us, and therefore the data set that we present all results on, are for the development set from the 1997 Hub4 NE task for which there are published results for two other systems [4]. The first, SPRACH-R, is a rule based system which has F measures of 69 and 59 on manual transcript and 1-best recogniser output respectively. The second, statistical system, SPRACH-S, does significantly better with F measures of 68 and 80 respectively. The 1-best transcript used in both cases had a WER of 27%.

5. Experiments

We present the results of a series experiments. In the first, we show that our system performs well on a manual transcript. In the second, a cheating experiment, we use the lowest WER path through the recogniser output word lattices. In the third, we use the 1-best path – this experiment corresponds to the conventional method for named entity extraction from speech. In the final experiment, we use our new method to maximise the joint probability of the entity and word sequences.

5.1. Manual transcript

On a manual transcript, our system has an F measure of 83.21. This confirms that our system performs at least as well as other systems on this task, such as SPRACH-S [4] which has an F measure of 80.

5.2. "Lattice best"

By reference to the manual transcript, the path through the lattice with the lowest possible WER can be found; the WER for the word string corresponding to this path is called the lattice error rate (LER) and is 5.4% for the lattices in our test set. We will refer to the word string along this path as the *lattice best*.

By running the system outlined in section 2 above on the *lattice best* word strings, we obtain an F measure of 78.87. This value can be seen as an upper bound on the performance of our system – noting however that a lower WER does not guarantee a higher F measure (see section 6).

5.3. 1-best

The word strings corresponding to the 1-best paths through the lattices have an overall WER of 19.7%. This path is the one that maximises the sum of the acoustic and recogniser language model log probabilities (suitable weighted by the language model scaling factor) and pays no attention to the named entities. The standard technique for named entity extraction from speech uses this word string, and by running our system on the 1-best word strings we obtain an F measure of 74.04. As we will see, the F measure can be increased by choosing an alternative path through the word lattice.

5.4. New method – full lattice

We used our new method to find the sequences E_1^L and W_1^L which maximise equation 1. The F measure can then be computed, as can the WER for the W_1^L found. The F measure was increased to 74.34 with the WER slightly worse at 20.1%.

Results are summarised in table 1 and figure 2. From the graph it is possible to see both the general trend that the higher the WER the lower the F measure, and also that by using word lattices rather than the 1-best transcription it is possible to move the operating point not along the usual operating curve but orthogonal to it (see section 6).

System	Input		F
	type	WER	
	manual transcript	0%	83.21
SPRACH-S	manual transcript	0%	80
	lattice best	5.4%	78.87
	full lattice	20.1%	**74.34**
	1-best word string	19.7%	74.04
SPRACH-R	manual transcript	0%	69
SPRACH-S	1-best word string	27%	68
SPRACH-R	1-best word string	27%	59

Table 1: *Results for various systems on the same data set. Systems named in italics are quoted from the literature for comparison, all other results are for our new system.*

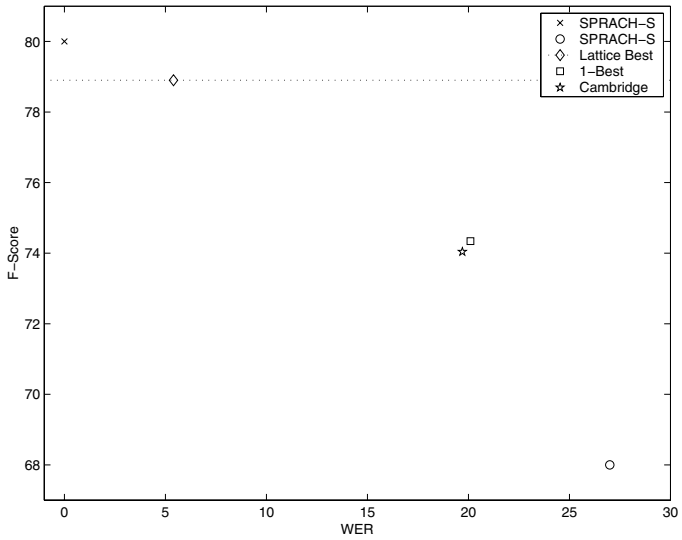

Figure 2: *Results from table 1*

6. Discussion

We have shown that it is possible to find paths through recogniser output word lattices that have a better F measure than the 1-best path. To do this, we maximised the joint probability of entity and word sequences, rather than the conditional probability of the entities *given* the word sequence. Although the improvement in F measure is modest, it clearly demonstrates that it is possible to improve F measures either by decreasing WER – moving in direction A along the operating curve shown in figure 3, or by trading off WER against F measure – moving in direction B – in which case the new operating point is one with improved F measure but worse word error rate. Moving in direction A requires improvements to the speech recogniser, whereas moving in direction B does not. We can envisage situations where trading off WER against F measure is acceptable – for example, systems which are primarily interested in finding named entities and less interested in accurate transcriptions of all words, such as audio search engines.

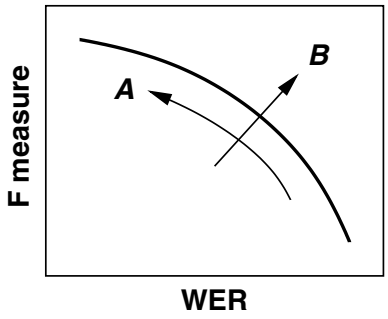

Figure 3: *Operating curve*

7. Conclusion

There is a general relationship between word error rate and F measure, as shown in figure 2. But, we have demonstrated that it is possible to gain some improvement in F measure at the expense of some WER – by moving orthogonal to the operating curve. We have shown that by using word lattices for named entity extraction, rather than the standard method of using the 1-best transcription, we have improved F measure.

We think that this improvement is not a result of the type of model that we have considered, but rather due to the fact that we use word lattices instead of 1-best transcriptions. We suggest that any method that does not assume that the word sequence is independent of the entity sequence will benefit from using word lattices.

8. Future work

A known weakness of our current system is that the language models in the states of the finite state machine are not trained discriminatively. We are currently investigating ways of making the models not only generate in-class word sequences with high probability, but generate out-of-class sequences with low probability. For example, the Place model should generate a word sequence tagged in the training data as a People entity with low probability. This must still be combined with some form of language model smoothing, because training data is sparse.

9. Acknowledgements

We would like to thank Phil Woodland of Cambridge University Engineering Department for providing the word lattices.

10. References

[1] D Bikel, S Miller, R Schwartz and R Weischedel *NYMBLE: A High-Performance Learning Name Finder* Proc. Applied Natural Language Processing 1997

[2] F Kubala, R Schwartz, R Stone and R Weishedel *Named Entity Extraction from Speech* Proc. Broadcast News Transcription and Understanding Workshop 1998

[3] D Palmer, J Burger and M Ostendorf *Information Extraction from Broadcast News Speech Data* Proc. Darpa Broadcast News Workshop 1999

[4] S Renals, Y Gotoh, R Gaizauskas and M Stevenson *Baseline IE-NE Experiments Using the Sprach/Lasie System* Proc. Darpa Broadcast News Workshop 1999

[5] H Kim *Named Entity Recognition form Speech and Its Use in the Generation of Enhanced Speech Recognition Output* PhD Thesis, Cambridge University 2001

[6] T Witten and I Bell *The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression.* IEEE Transactions on Information Theory 1991

[7] Linguistic Data Consortium 1998 *Catalogue references LDC98E10 & LDC98E11* www.ldc.upenn.edu

[8] S. J. Young and N. H. Russell and J. H. S. Thornton *Token passing : a simple conceptual model for connected speech recognition.* 1989 Cambridge University Engineering Department technical report CUED/F-INFENG/TR.38

[9] P.R. Clarkson and R. Rosenfeld *Statistical Language Modeling Using the CMU-Cambridge Toolkit* 1997 Proceedings Eurospeech 1997