FUZZY CLASS RESCORING: A PART-OF-SPEECH LANGUAGE MODEL

P. Geutner

pgeutner@ira.uka.de

Interactive Systems Laboratories
Department of Computer Science,
University of Karlsruhe,
76128 Karlsruhe, Germany

ABSTRACT

Current speech recognition systems usually use word-based trigram language models. More elaborate models are applied to word lattices or N best lists in a rescoring pass following the acoustic decoding process. In this paper we consider techniques for dealing with class-based language models in the lattice rescoring framework of our JANUS large vocabulary speech recognizer. We demonstrate how to interpolate with a Part-of-Speech (POS) tag-based language model as example of a class-based model, where a word can be member of many different classes. Here the actual class membership of a word in the lattice becomes a hidden event of the A^* algorithm used for rescoring. A forward type of algorithm is defined as extension of the lattice rescorer to handle these hidden events in a mathematically sound fashion. Applying the mixture of viterbi and forward kind of rescoring procedure to the German Spontaneous Scheduling Task (GSST) yields some improvement in word accuracy. Above all, the rescoring procedure enables usage of any fuzzy/stochastic class definition for recognition units that might be determined through automatic clustering algorithms in the future.

1. INTRODUCTION

Speech recognition systems based on words as elementary recognition units usually use trigrams during search within the primary acoustic decoding process. More elaborate statistical language models are then applied later on using word lattice rescoring techniques to extract the first best hypothesis from lattices produced during recognition runs.

Application of class-based language models is one way of rescoring the generated word lattice [1]. Using predefined classes (e.g function and content words [2]) is one option explored earlier. Also, automatically created (using optimization criteria like the amount of perplexity reduction) classes were proposed and evaluated [3].

One thing these clustering algorithms and hand-made word classifications have in common is, that each word is assigned to exactly one class. That means, given a word sequence $w = (w_1, w_2, \ldots, w_n)$ and the inital language model $p(w)$, the class-based linearly interpolated language model $p'(w)$ can be derived from the probabilities $p(w_i|c_i)$ of observing word w_i given a class c_i and the class language model $p(c)$ in the following straightforward way

$$\begin{aligned} p'(w) &= \lambda p(w) + (1-\lambda) p(w|c) p(c) \\ &= \lambda p(w) + (1-\lambda) \prod_{i=1}^{n} p(w_i|c_i) p(c_i|c_{i-1}) \end{aligned}$$

assuming for the sake of simplicity that $p(c)$ is a bigram language model.

Besides, there is also the possibility of stochastic/fuzzy classes where each word is not uniquely assigned to one single class, but can be member of different classes with varying probabilities. One example shown in this paper is the use of Part-of-Speech (POS) classes [4]. Here a number of words has no unique class (= part-of-speech tag), but different tags can be potentially assigned to a word depending on the grammatical role of the word in a certain context. In the following we will report on how the rescoring of those stochastic class models can be accomplished in an efficient way.

2. PART-OF-SPEECH TAGS

All experiments done for this research were performed on the **G**erman **S**pontaneous **S**cheduling **T**ask (GSST). This database consists of human-to-human dialogues where two individuals are given different calendars with various appointments. Goal of the conversations is to schedule a meeting.

The training corpus was tagged using a Part-of-Speech Tagger provided by the University of Stuttgart [5] which is based on Hidden Markov Models and could be easily applied to our German database. The tagger is reported to work with an accuracy of 96%.

Based on the tagged training corpus we computed the lists of classes a word w_i can belong to and the probabilities $p(w_i|c_i)$. Naturally many words have unambigious assignments to their classes, especially this is the case for names like "Hamburg", "John" or nouns like "April", "Treffen" (= meeting) and so on. Still there is enough variation to yield an improvement in word accuracy when taking advantage of the fuzzy class definition of POS tags.

So for example the word "überlegen" (= think about) can be a finite verb or an infinitive verb:

- "Wir überlegen, ob wir nach Hause gehen." (= we think about going home) FINITE VERB
- "Wir müssen überlegen." (= we have to think about it) INFINITIVE VERB

Similarly "vor" (= in front of) can be a preposition or verb prefix:

- "Ich bin vor dem Hotel." (= I am in front of the hotel) PREPOSITION
- "Ich bereite das vor." (= I prepare this) VERB PREFIX

Number of Tags	Number of Words
1	5050
2	724
3	118
4	40
5	13
6	15
7	4
8	2

Table 1. Tag Distribution: Number of alternative tags per word to indicate the fuzzy nature of the part-of-speech tags.

Table 1 lists the number of words that can be tagged with one to eight different class tags. The majority of words have a deterministic tag assignment (e.g. nouns and names), but about 15% of all vocabulary words can be mapped to various tags with different probabilities.

3. LANGUAGE MODELS IN JRTK

The Karlsruhe-Verbmobil Speech Recognition Engine is based on the Janus Speech Recognition Toolkit (JRTk) developed at the Interactive Systems Laboratories in Karlsruhe and at Carnegie Mellon University in Pittsburgh[6, 7]. This toolkit implements a new object-oriented approach. A flexible Tcl/Tk script based environment allows building state-of-the-art multimodal recognizers – this includes speech, handwriting and gesture recognition. Unlike other toolkits Janus is not a set of libraries and precompiled modules but a programmable shell with transparent, yet very efficient objects.

The Toolkit allows for many different types of language model objects to interact with each other. What all language model objects have in common is a method to compute the conditional probability $p(w_n \dots w_{n-k} | w_{n-k-1} \dots w_0)$. Among those types are

LModel This is the basic language model object that handles n-gram based language models within Janus.

LModelMap A mapping language model is defined through a list of "word w is mapped from word v with probability $p(w|v)$". The conditional probability $p(w_n \dots w_{n-k} | w_{n-k-1} \dots w_0)$ is computed by mapping the sequence of w's to a sequence of v's and multiplying the mapping probability $p(w|v)$ with the probability $p(v_n \dots v_{n-k} | v_{n-k-1} \dots v_0)$ computed within another language model object.

LModelInt This language model object type implements a linear interpolation scheme between two different JRTk language model objects.

Based on this object-oriented architecture the Janus Speech Recognition Toolkit allows for a hierarchy of language models. Consider for example implementing a part-of-speech-based language model interpolated with a word-based language model. Figure 1 shows the hierarchical structure as used for this paper.

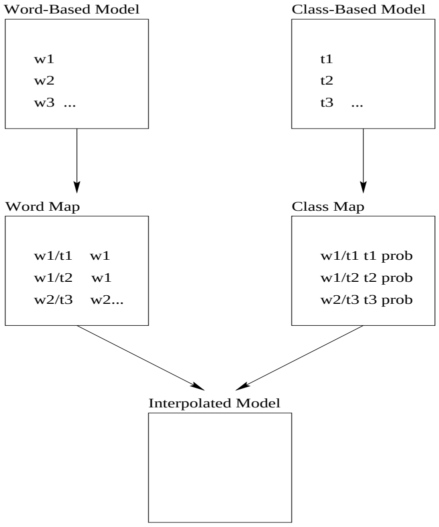


Figure 1. Class Hierarchy of language models within JRTk to implement a POS/word-based interpolated language model object.

The two underlying models are a word-based trigram model where trigram sequences of the words w_1, w_2, …, w_n in the vocabulary are modeled. The second model consists of a class-based trigram model (here the classes are POS tags), where tag sequences over the tag vocabulary t_1, t_2, …, t_m are modeled. For both models there exists a map model underneath that is able to map word/tag combinations to either the word (for the word model) or the tag (for the POS model). For the word model the word/tag → word mapping is non-ambiguous, e.g. $w_1/t_1 \to w_1$, $w_2/t_2 \to w_2$. In the latter case the word/tag → tag mapping is annotated with a certain probability, as every word can belong to several tag classes: e.g. $w_1/t_1 \to t_1$ with probability 0.5, $w_1/t_2 \to t_2$ with probability 0.4.

It is only through these mappings that the POS/word-based interpolated LM can deal with tagged word sequences $w_1/t_1\ w_2/t_2 \dots w_k/t_k$, and is able to make use of the underlying word- and class-based models.

The overall implementation of this hierarchical approach allows for easy and convenient testing of many different language models that might be combined the same way as the two models in our example. Other examples are language models implementing hidden utterance segment boundaries [7].

4. RESCORING ALGORITHM

Since class membership of words is no longer unique, the class variable c_i becomes a hidden variable. Thus, we have to sum over all possible sequences of class assignments $c = (c_1, c_2, \dots, c_n)$ for a string of words $w = (w_1, w_2, \dots, w_n)$

$$\begin{aligned} p'(w) &= \lambda p(w) + (1-\lambda) \sum_c p(w|c) p(c) \\ &= \lambda p(w) + (1-\lambda) \sum_c \prod_{i=1}^{n} p(w_i|c_i) p(c_i|c_{i-1}) \end{aligned}$$

In order to do this summation in an efficient way within the lattice, we implemented a dynamic programming approach for computing the sums over the different class sequence hypotheses.

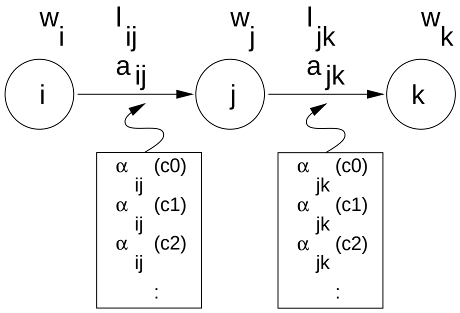
Let L be the word lattice consisting of vertices w_i and directed edges e_{ij} pointing from w_i to w_j, where w_0 is supposed to be the unique start node and w_N the unique end node. Each edge e_{ij} is labelled with the acoustic log likelihood a_{ij} of observing the word w_i preceding word w_j, the log likelihood l_{ij} of the best path through the lattice up to the node w_j so far (initialized by $-\infty$) and a pointer p_{ij} to the preceding edge of the best path.

4.1. Viterbi Rescoring

A very simple and straightforward approach to rescoring is to expand the lattice of the recognition run such that we replace each word node w by all possible word/tag nodes $w/t_1 \ldots w/t_k$. Based on the expanded lattice we can do rescoring with respect to our hierarchical language model without any modification required for the rescoring procedure. The potential drawback of this approach is, that we consider the most likely tag sequence for a given word sequence and not the sum over all possible sequences as stated above.

4.2. Forward Rescoring

Let $\alpha_{ij}(c_m)$ denote the normalized probability of observing the words on the best path leading to arc e_{ij} and ending up in class c_m for word w_j based on the class-based language model.



The lattice is traversed in an A^* like fashion. That means, for each edge e_{0i} originating from the start node, we compute the normalized forward probabilities $\alpha_{0i}(c_l)$ and the log likelihoods l_{0i} as

$$\begin{aligned} \alpha_{0i}(c_l) &= \frac{p(w_i|c_l)\; p(c_l|c_{start})}{\underbrace{\sum_{c_n} p(w_i|c_n)\; p(c_n|c_{start})}_{=s_{0i}}} \\ l_{0i} &= a_{0i} + \lambda \log p(w_i|w_0) + (1-\lambda)\log s_{0i} \end{aligned}$$

and add each of the edges to a queue. Then, as long as the queue is not empty, we pick the element e_{ij} from top of the queue and compute for each edge e_{jk} the following update:

1. For each class c_l word w_k is a member of, we compute the normalized forward probability

$$\alpha(c_l) = \frac{p(w_k|c_l)\sum_{c_m}\alpha_{ij}(c_m)\; p(c_l|c_m)}{\underbrace{\sum_{c_n} p(w_k|c_n)\sum_{c_m}\alpha_{ij}(c_m)\; p(c_n|c_m)}_{=s_{jk}}}$$

2. Compute the combined log-likelihood of the acoustics, the word-based language model and the fuzzy class-based language model as

$$l = l_{ij} + a_{jk} + \lambda \log p(w_k|w_i, w_j) + (1-\lambda)\log s_{jk}$$

   This equation makes use of the fact that the scaling factors s_{jk} are sufficient to compute the probability of an observation sequence given an HMM.[1]

3. If the combined likelihood l turns out to be larger than the likelihood l_{jk} attached to that edge, then redefine the following variables

$$\begin{aligned} l_{jk} &= l \\ p_{jk} &= e_{ij} \\ \alpha_{jk}(c_l) &= \alpha(c_l) \end{aligned}$$

   This way we keep track of the best path going through e_{jk} as found so far.

4. If there is no other arc $e_{\imath j}$ left in the queue add e_{jk} to the queue.

5. RESULTS

All recognition results reported have been performed with the JANUS system [8]. The training set for training the word-based trigram model consisted of 330.000 words, the same text was used to build a POS language model. An independent test set consisting of 35 GSST dialogues containing 6.300 words has been defined and word accuracy results are reported on this set. All experiments performed are summarized in Table 2.

Baseline experiment is a recognition run with a conventional word-based trigram language model which resulted in a 85.1% word recognition rate. The lattice output of this baseline recognition run was used as starting point for the following suite of experiments for a fixed, context independent interpolation weight. Table 2 shows that the viterbi rescoring performs worse than the baseline whereas the fuzzy rescoring approach yields a small gain in performance.

	Word Accuracy
Word Trigram Model	85.1% WA
Interpolation of Word Trigram Model and POS Model (Viterbi Rescoring)	84.7% WA
Interpolation of Word Trigram Model and POS Model (Forward Rescoring)	85.3% WA

Table 2. Recognition Results

6. CONCLUSIONS

It has been shown that the use of grammatical and linguistic knowledge hidden in POS tags can be used to improve recognition results. The main difficulty of using a POS class division is the ambiguity of class affiliations. Whereas usually POS language models are used to find the most probable tagging sequence for a certain sequence of words, here a new rescoring algorithm has been applied. This algorithm is capable of dealing with fuzzy classes in a mathematically sound way by considering all possible tag assignments for a certain word.

[1] The proof of this is out of scope of this paper. It makes use of the observation, that the normalized forward probabilities are proportional to the unnormalized ones times the product of all normalization factors applied.

Currently linear interpolation is used to combine the word-based trigram language model with the POS model. Through the hierarchical implementation described above many different language models can be combined into one single model very easily.
Finally, the presented rescoring algorithm allows to use any class definition of words where no unique class can be assigned. Other automatic clustering procedures than the conventional ones can then be used to assign words to several classes with different probabilities.

7. ACKNOWLEDGEMENTS

This research was partly funded by grant 413-4001-01IV101S3 from the German Federal Ministry of Education, Science, Research and Technology (BMBF) as a part of the VERBMOBIL project. The views and conclusions contained in this document are those of the author. The author whishes to thank Helmut Schmid for making his POS-Tagger publicly available.

REFERENCES

[1] A. Farhat, J.-F. Isabelle and D. O'Shaughnessy. *Clustering Words for Statistical Language Models Based on Contextual Word Similarity.* Proceedings of the IEEE 1996 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Atlanta, Georgia, pp. 180-183, May 1996.

[2] P. Geutner. *Introducing Linguistic Constraints into Statistical Language Modeling.* ICSLP'96, Philadelphia, Pennsylvania, October 1996.

[3] R. Kneser and H. Ney. *Improved Clustering Techniques for Class-Based Statistical Language Modelling.* 3rd European Conference on Speech Communication and Technology (EUROSPEECH), Berlin, Germany, pp. 973-976, September 1993.

[4] T.R. Niesler and P.C. Woodland. *Variable-length category-based n-grams for language modelling.* Technical Report, Cambridge University, UK, April 1995.

[5] H. Schmid. *Improvements in Part-of-Speech Tagging with an Application to German.* EACL SIGDAT Workshop, Dublin, Ireland, 1995.

[6] M. Finke, P. Geutner, H. Hild, T. Kemp K. Ries and M. Westphal. *The Karlsruhe-Verbmobil Speech Recognition Engine.* Proceedings of the IEEE 1997 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Munich, Germany, May 1997.

[7] M. Finke, J. Fritsch, P. Geutner, K. Ries, T. Zeppenfeld and A. Waibel *The JanusRTk Switchboard/Callhome 1997 Evaluation System.* Proceedings of the LVCSR Hub-5 Workshop, Baltimore, Maryland, May 1997.

[8] P. Geutner et al.'*Integrating Different Learning Approaches into a Multilingual Spoken Language Translation System.* In: *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing.* Lecture Notes in Artificial Intelligence, S. Wermter, E. Riloff and G. Scheler (Eds.), pp. 117-131, Springer, 1996.