Using Boosting and POS Word Graph Tagging to Improve Speech Recognition

Christer Samuelsson[1] and James L. Hieronymus[2]

[1]Xerox Research Centre Europe
6, chemin de Maupertuis
38240 Meylan, FRANCE
Christer.Samuelsson@xrce.xerox.com

[2]RIACS, M/S T27A–2
NASA Ames Research Center/S: T27A-2
Moffett Field, CA 94035-1000
jimh@riacs.edu

Abstract

The word graphs produced by a large vocabulary speech recognition system usually contain a path labelled with the correct utterance, but this is not always the highest scoring path. Boosting increases the probability of words which occur often in the word graph, which are in some sense robust. Adding syntactic information allows rescoring of arc probabilities with the possibility that more grammarical word sequences will also be the correct ones. A theory is developed which allows general probabilistic syntactic models to be used to rescore word lattices. Experiments conducted on the Wall Street Journal (WSJ) corpus with a version of the AT&T 1995 FST LVSR system with part of speech (POS) trigram sequences show that using only POS leads to a loss in performance. Boosting alone provides an improvement in performance which is not statistically significant. Cascading the two methods, boosting first and then using syntactic information improves performance 4.5 % relative on a large portion of the 1995 DARPA test set.

1. Introduction

There have been several attempts to improve LVSR performance using grammatical and categorical information. Usually the categorical n-gram techniques work best, by providing back off's for novel word sequences not seen in the training data. The performance increases have been relatively modest. Kneser and Ney ([Kneser & Ney 1993]) used automatically derived categories (ADCAT) from data to improve speech recognition. Niesler et al ([Niesler *et al* 1998]) compared POS and ADCAT n-grams for speech recognition and found a small improvement (for POS n-grams) and approximately 7% relative improvements for ADCAT n-grams on 1994 WSJ dev and test sets.

This paper details two approaches to word graph rescoring to improve speech recognition performance. The first approach uses boosting, that is favoring a word which occurs frequently in the graph by boosting its probability by the number of occurances. Word graphs often have multiple occurances of the correct word due to small differences in the start and end times.

The second approach attempts to incorporate higher-level syntactic information, hopefully penalizing ungrammatical paths. This approach is quite general and applicable to a large number of stochastic grammars, such as tag N-gram models and stochastic context-free grammars, [Booth & Thompson 1973]. The only restriction is that the syntactic information must be mediated by part-of-speech tags, which are syntactic labels assigned to each word.

The rest of the article is organized as follows: In Section 2 we discuss how to filter out spurious words by weighting the arcs with the global frequencies of the words. In Section 3 we describe the proposed method for rescoring the word graphs using syntactic information. The mathematical details of the two approaches are placed in Section 4. In Section 5 we present the results of an empirical evaluation of the two approaches, both in isolation and cascaded.

2. Frequency-based Filtering

Frequency-based filtering relies on the observations that in the word graphs, the same word is hypothesized many times with slight variations in start and end times. It is actually possible to count the total fractional number of occurrences of each word in a word graph by multiplying each arc probability with the probability of its start node, and summing over all arcs labeled with the same word. We then simply multiply the probability of each arc in the word graph with the total frequency of occurrences of its word label in the graph. The arc probabilities are then redistributed in the graph using an operation known as pushing [Mohri 1997], rather than simply renormalized. The mathematical details of this approach are given in Section 4.

3. Syntactic Rescoring

The only restriction on the syntactic information used to rescore the word graph is that it must be mediated by part-of-speech (PoS) tags, which are syntactic labels assigned to each word. First, the word graph is transduced to a PoS tag graph. Each word is looked up in a lexicon, and output arcs labeled with each possible tag are generated and assigned the probability

$$P(W,T \mid G) \quad = \quad P(T \mid W) \cdot P(W \mid G)$$

Here $P(T \mid W)$ comes from the tag lexicon, and $P(W \mid G)$ from the original word graph. Identical tag arcs are summed over:

$$P(T \mid G) \quad = \quad \sum_W P(W,T \mid G)$$

Next, the tag graph is rescored using some model of syntax S:

$$P(T \mid G) \quad \Rightarrow \quad P(T \mid G,S)$$

Then, the original word graph is rescored using the new and old tag-graph probabilities $P(T \mid G,S)$ and $P(T \mid G)$:

$$P(W \mid G,S) \quad = \quad \sum_T P(W,T \mid G) \cdot \frac{P(T \mid G,S)}{P(T \mid G)}$$

This formula is derived in rigorous detail in Section 4.

The formula suggests a very clear-cut architecture for realizing this approach computationally: The arcs of the word

graph are first transduced to arcs labeled with word-tag pairs. These are used both as input to the rescoring module and to construct the tag arcs by summing over words. The latter will also serve as input to the rescoring module, as well as be rescored using some, here unspecified, stochastic model of syntax, and the rescored tag arcs will constitute the third input stream of the rescoring module. These three arc streams are combined according to the formula above, which gives us the rescored word graph.

We realize that, despite the minor overhead of synchronizing the three input streams of the rescoring module, this architecture will allow incremental real-time processing if the speech recognizer and the syntactic parser do so.

4. Mathematical Details

We here derive in mathematical detail the formulas used in the two rescoring approaches. Some readers may wish to skip this section.

4.1. Labeled probabilistic graphs

A labeled probabilistic graph G consists of a set of nodes $\mathbf{N} = \{N_0, N_1, \ldots, N_n\}$, where N_0 is a distinguished initial node; a set of labels $\mathbf{L} = \{L_1, \ldots, L_m\}$; and a set of directed labeled arcs $\mathbf{A} = \{\langle N_i, L_k, N_j\rangle\}$, i.e., a subset of $\mathbf{N}\times\mathbf{L}\times\mathbf{N}$. A probability distribution $P(\langle N_i, L_k, N_j\rangle \mid N_i)$ over the set of outgoing arcs is associated with each node N_i; for each node, the sum of the probabilities over the set of outgoing arcs equals one. We will require that the graphs are acyclic, i.e., that there is no arc sequence

$$\langle N_{i_1}, L_{k_1}, N_{i_2}\rangle, \ldots, \langle N_{i_{t-1}}, L_{k_{t-1}}, N_{i_t}\rangle$$

such that $N_{i_1} = N_{i_t}$.

4.2. Node and label probabilities

For each node we can calculate the probability of visiting it, given that we start out in the initial node N_0:

$$P(N_j \mid G) = \\ = \begin{cases} 1 & j = 0 \\ \sum_{i,k} P(N_i \mid G)\cdot P(\langle N_i, L_k, N_j\rangle \mid N_i, G) & j \neq 0 \end{cases}$$

We can thus calculate the total fractional number of occurrences of any label in the graph by multiplying each arc probability with the node probability of its start node, and summing over all arcs with identical labels:

$$P(L_k \mid G) = \\ = \sum_{\langle N_i, L_k, N_j\rangle \in \mathbf{A}} P(N_i \mid G)\cdot P(\langle N_i, L_k, N_j\rangle \mid N_i, G)$$

4.3. Frequency weighting

The filtering approach consists in multiplying the probability of each arc in the word graph produced by the speech recognizer with the total number of occurrences in the graph of the word it is labeled with:

$$P'(\langle N_i, W, N_j\rangle \mid N_i, G) = \\ = P(W \mid G)\cdot P(\langle N_i, W, N_j\rangle \mid N_i, G)$$

The arc probabilities are then redistributed in the graph using an operation known as pushing, see [Mohri 1997], rather than through simple renormalization. There is little theoretical justification for this approach, but it seems to work well in practice.

4.4. Word-to-tag transduction

We can transduce a probabilistic word graph $\langle N_i, W, N_j\rangle$ to a probabilistic tag graph $\langle N_i, T, N_j\rangle$, using a set of transduction rules, i.e., the tag lexicon:

$$W \rightarrow T$$

The arc probabilities are related by the equation

$$P(\langle N_i, T, N_j\rangle \mid N_i, G) = \\ = \sum_W P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G) \quad (1)$$

which in turn relies on the lexical tag probabilities $P(T \mid W)$ and the equation

$$P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G) = \\ = P(T \mid W)\cdot P(\langle N_i, W, N_j\rangle \mid N_i, G) \quad (2)$$

Here the conditioning on G indicates using the probability distributions of the original word graph.

4.5. The rescoring formula

Let us use some syntactic information S to rescore the arc probabilities of the tag graph, which we will indicate by conditioning them on S, and then use these to rescore the arc probabilities of the word graph, thus incorporating the syntactic information S:

$$P(\langle N_i, W, N_j\rangle \mid N_i, G, S) = \\ = \sum_T P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G, S) \quad (3)$$

where we have summed over all possible tags T assignable to W. Using the definition of conditional probability

$$P(A \mid B) = \frac{P(A, B)}{P(B)}$$

we get

$$P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G, S) = \\ = P(\langle N_i, W, N_j\rangle \mid \langle N_i, T, N_j\rangle, G, S)\cdot \\ \cdot P(\langle N_i, T, N_j\rangle \mid N_i, G, S) \quad (4)$$

The syntactic information S is assumed to be mediated by the part-of-speech tags introduced by the tag lexicon. This is consistent with for example the basic assumptions underlying part-of-speech tagging using hidden Markov models, [Church 1988]. This means that once $\langle N_i, T, N_j\rangle$ is specified, it is assumed that no further syntactic information affects the word probability from N_i to N_j, and we thus have

$$P(\langle N_i, W, N_j\rangle \mid \langle N_i, T, N_j\rangle, N_i, G, S) = \\ = P(\langle N_i, W, N_j\rangle \mid \langle N_i, T, N_j\rangle, N_i, G) \quad (5)$$

We again use the definition of conditional probability

$$P(\langle N_i, W, N_j\rangle \mid \langle N_i, T, N_j\rangle, N_i, G) = \\ = \frac{P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G)}{P(\langle N_i, T, N_j\rangle \mid N_i, G)} \quad (6)$$

Assembling Eqs. (4)–(6) we find that

$$P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G, S) = \quad (7)$$
$$= P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G) \cdot$$
$$\cdot \frac{P(\langle N_i, T, N_j\rangle \mid N_i, G, S)}{P(\langle N_i, T, N_j\rangle \mid N_i, G)}$$

and inserting this into Eq. (3) yields

$$P(\langle N_i, W, N_j\rangle \mid N_i, G, S) = \quad (8)$$
$$= \sum_T P(\langle N_i, W, N_j\rangle, \langle N_i, T, N_j\rangle \mid N_i, G) \cdot$$
$$\cdot \frac{P(\langle N_i, T, N_j\rangle \mid N_i, G, S)}{P(\langle N_i, T, N_j\rangle \mid N_i, G)}$$

which is the syntactic rescoring formula given in Section 3, in slight notational disguise.

5. Experiments

The two approaches were applied to a collection of about 180 word graphs from the 1995 DARPA evaluation test set in the NAB domain, generated by a large-vocabulary FST based speech recognizer, developed at Bell Labs [Ljolje *et al* 1995]. Speaker normalization was not used in this case, so the word error rates are around 22%. The speech recognizer employed a backed-off word bigram model extracted from a 40 million word corpus in the WSJ and NAB domains. The syntactic model used for rescoring consisted of a smoothed tag trigram model and lexical tag probabilities extracted from the 1 million word WSJ portion of the Penn Treebank II. The lexical tag probabilities were rescored under the tag trigram model as described in [Samuelsson 1997a, Samuelsson 1997b].

Frequency-based filtering decreased the recognition error-rate by 2.8% relative, which is not statistically significant. One potential problem with this approach is that it allows hypothesized words at radically different times to interact. Windowing, i.e., only counting arcs within a certain time distance from the arc being rescored, alleviates this problem but ignores self-triggering effects. Windowing did not change the results.

Syntactic rescoring changed the top hypothesis in many cases, usually to more grammatical constructions. For example, "more cheaply and options" was changed to "more cheaply than options" which was correct. It introduced slightly more errors than it corrected, resulting in 3.3% relative increase in error rate. We attribute this somewhat disappointing result to the similarity between the word bigram and tag trigram models.

The most interesting experimental result was however cascading the two approaches, first filtering on frequency and then rescoring based on syntactic information; this did reduce the error rate substantially, by 4.5% relative, which is statistically significant. The table of Figure 1 summerizes the results. It appears to bring down the noise level in the word graph to a level were syntactic rescoring can be effective.

	Error rate	Rel. change
Original system	816/3541	—
Syntactic rescoring	843/3541	$+3.3\%$
Frequency-based filtering	793/3541	-2.8%
Rescoring and filtering	779/3541	-4.5%

Figure 1: Experimental results.

The experiments also verified the computational feasibility of the approach, as it allowed real-time incremental rescoring — processing times were totally dominated by speech recognition.

6. Discussion

The proposed approach for syntactic rescoring allows feeding back syntactic information into the speech recognition process in a very clear-cut way, with well-defined and natural interfaces between the speech recognizer and the syntactic parser. The approach is applicable to a large class of stochastic models of syntax, namely those where the syntactic information is mediated through part-of-speech tags. These include various extensions to stochastic context-free grammars, for example [Collins 1997] and [Manning & Carpenter 1997].

The work by Niesler ([Niesler *et al* 1998]) used variable length n-grams of POS tags as a backoff mechanism. His results show that the best POS n-grams were 4 and 5 grams, while trigrams were 1/3 as effective. This suggests that the present work should be extended to higher order n-grams. The present results have obtained a 4.5 % decrease in word error, and the Niesler technique 7 %. So it is possible that the cascade technique on POS 4-grams will produce better results. The present result uses POS sequences for both seen and unseen n-grams, and thus provides a potential gain in gramaticality even for word trigrams, which could be the result of concatinating incompatible POS sequences. There have been attempts to sort hypotheses from the speech recognizer and rescore them according to subsequently applied models of syntax, for example [Sankar *et al* 1996] (part-of-speech tags), [Sekine *et al* 1997] (SCFGs) and [Moore *et al* 1995, Rayner *et al* 1994] (G/HPSG-style unification grammars). The last one comes the closest to operating on the word graph produced by the speech recognizer by recompiling the N-best hypotheses into a word graph, which is subsequently analyzed. In all these cases, however, the hypothesis corresponding to the best analyses (including acoustic scores) is selected, rather than the globally most likely hypothesis. Part of speech backoffs from word n-grams using overlapping POS categories were investigated by Samuelsson and Reichl ([Samuelsson & Reichl 1999]) on the WSJ 1994 test set using a single pass recognizer. They report an improvement of 5 % in relative word error rate. Thus showing the value of POS backoffs for improving recognition.

7. Acknowledgements

This work was begun at Bell Labs. Discussions with B. Carpenter, J. Chu-Carroll, A. Ljolje, M. Riley, D. Hindle and M. Mohri were helpful in this work.

8. References

[Booth & Thompson 1973] T. L. Booth and R. A. Thompson. 1973. "Applying Probability Measures to Abstract Languages". In *IEEE Transactions on Computers, C-22(5)*, pp. 442–450.

[Church 1988] K. W. Church. 1988. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text". In *Procs. 2nd Conference on Applied Natural Language Processing*, pp. 136–143, ACL, 1988.

[Collins 1997] Michael Collins. 1997. "Three Generative, Lexicalized Models for Statistical Parsing". In *Procs. 35th Annual Meeting of the Association for Computational Linguistics*, pp. 16–23, ACL, 1997.

[Kneser & Ney 1993] R. Kneser and H. Ney. 1993. “Improved cludstering techniques for class based statistical language modelling”. In *Proc. Eurospeech-93*, pp. 973–976, 1993.

[Ljolje *et al* 1995] Andrej Ljolje, Michael Riley, Donald Hindle and Fernando Pereira. 1995. “The AT&T 60,000 Word Speech-To-Text System”. In *Procs. Spoken Language Systems Technology Workshop (ARPA)*, pp. 162–164, Austin, January 1995. Morgan Kaufmann.

[Manning & Carpenter 1997] Christopher Manning and Bob Carpenter. 1997. “Probabilistic Left Corner Grammars”. In *Procs. 5th International Workshop on Parsing Technologies*, Boston, Massachusetts, USA.

[Mohri 1997] Mehryar Mohri. 1997. “Finite-State Transducers in Language and Speech Processing”. In *Computational Linguistics 23(2)*, pp. 269–312, The MIT Press.

[Moore *et al* 1995] Robert Moore, Douglas Appelt, John Dowding, J. Mark Gawron and Douglas Moran. 1995. “Combining Linguistic and Statistical Knowledge in Natural-Language Processing for ATIS”. In *Procs. Spoken Language Systems Technology Workshop (ARPA)*, pp. 261–264, Austin, January 1995. Morgan Kaufmann.

[Rayner *et al* 1994] Manny Rayner, David M. Carter, Vassilios V. Digalakis and Patti Price. 1994. “Combining Knowledge Sources to Reorder N-Best Speech Hypothesis Lists”. In *ARPA (HLT) Proceedings*, Princeton. Also available as Report CRC-044, SRI International, Cambridge, England.

[Niesler *et al* 1998] T R. Niesler, E. W. D. Whittaker and P. C. Woodland. 1998. “Comparision of part-of-speech and automatically derived category-based language models for speech recognition”. In *Proc. ICASSP-98*, 1998.

[Samuelsson 1997a] Christer Samuelsson. 1997. “Extending N-gram Tagging to Word Graphs”. In *Procs. 2nd International Conference on Recent Advances in Natural Language Processing*, pp. 21–26, Tzigov Chark, Bulgaria.

[Samuelsson 1997b] Christer Samuelsson. 1997. “A Left-to-right Tagger for Word Graphs”. In *Procs. 5th International Workshop on Parsing Technologies*, pp. 171–176, ACL.

[Samuelsson & Reichl 1999] Christer Samuelsson and Wolfgang Reichl. 1999. “A Class-based Language Model for Large-vocabulary Speech Recognition Extracted from Part-of-Speech Statistics”. In *Procs. ICASSP99, IEEE*.

[Sankar *et al* 1996] A. Sankar, A. Stolcke, T. Chung, L. Neumeyer, M. Weintraub, H. Franco and F. Beaufays. 1996. “Noise-resistant Feature Extraction and Model Training for Robust Speech Recognition”. In *Procs. Speech Recognition Workshop (DARPA)*. February 1996.

[Sekine *et al* 1997] Satoshi Sekine, Andrew Borthwick, Ralph Grisman. 1997. “NYU Language Modeling Experiment for 1996 CSR Evaluation” *Procs. Speech Recognition Workshop (DARPA)*, pp. 123–128. February 1997.