SPEECH RECOGNITION AND THE FREQUENCY OF RECENTLY USED WORDS : A MODIFIED MARKOV MODEL FOR NATURAL LANGUAGE

Roland Kuhn
School of Computer Science, McGill University
805 Sherbrooke St. West, Montreal

Abstract

Speech recognition systems incorporate a language model which, at each stage of the recognition task, assigns a probability of occurrence to each word in the vocabulary. A class of Markov language models identified by Jelinek has achieved considerable success in this domain. A modification of the Markov approach, which assigns higher probabilities to recently used words, is proposed and tested against a pure Markov model. Parameter calculation and comparison of the two models both involve use of the LOB Corpus of tagged modern English.

1 Introduction

Speech recognition systems consist of two components. An acoustic component matches the most recent acoustic input to words in its vocabulary, producing a list of the most plausible word candidates together with a probability for each. The second component, which incorporates a language model, utilizes the string of previously identified words to estimate for each word in the vocabulary the probability that it will occur next. Each word candidate originally selected by the acoustic component is thus associated with two probabilities, the first based on its resemblance to the observed signal and the second based on the linguistic plausibility of that word occurring immediately after the previously recognized words. Multiplication of these two probabilities produces an overall probability for each word candidate.

Our work focuses on the language model incorporated in the second component. The language model we use is based on a class of Markov models identified by Jelinek, the "n-gram" and "Mg-gram" models [Jelinek 1985, 1983]. These models, whose parameters are calculated from a large training text, produce a reasonable non-zero probability for every word in the vocabulary during every stage of the speech recognition task. Our model incorporates both a Markov 3g-gram component and an added "cache" component which tracks short-term fluctuations in word frequency.

We adopted the hypothesis that a word used in the recent past is much more likely to be used soon than either its overall frequency in the language or a Markov model would suggest. The cache component of our model estimates the probability of a word from its recent frequency of use. The overall model uses a weighted average of the Markov and cache components in calculating word probabilities, where the relative weights assigned to each component depend on the part of speech (POS).

For each POS, the overall model may therefore place more reliance on the cache component than on the Markov component, or vice versa; the relative weights are obtained empirically for each POS from a training text. This dependance on POS arises from the hypothesis that a content word, such as a particular noun or verb, will occur in bursts. Function words, on the other hand, would be spread more evenly across a text or a conversation; their short-term frequencies of use would vary less dramatically from their long-term frequencies. One of the aims of our research was to assess this hypothesis empirically. If it is correct, the relative weight calculated from the training text for the cache component for most content POSs will be higher than the cache weighting for most function POSs.

We intend to compare the performance of a standard 3g-gram Markov model with that of our model (containing the same Markov model along with a cache component) in calculating the probability of 100 texts, each approximately 2000 words long. The texts are taken from the Lancaster-Oslo/Bergen (LOB) Corpus of modern English [Johansson et al 1986, 1982]; the rest of the corpus is utilized as a training text which determines the parameters of both models. Comparison of the two sets of probabilities will allow one to assess the extent of improvement over the pure Markov model achieved by adding a cache component. Furthermore, the relative weights calculated from the training text for the two components of the combined model indicate those POSs for which short-term frequencies of word use differ drastically from long-term frequencies, and those for which word frequencies stay nearly constant over time.

2 A Natural Language Model with Markov and Cache Components

The "trigram " Markov language model for speech recognition developed by F. Jelinek and his colleagues uses the context provided by the two preceding words to estimate the probability that the word W_i occurring at time i is a given vocabulary item W. Assume recursively that at time i we have just recognized the word sequence $W_0, \cdots, W_{i-2}, W_{i-1}$. The trigram model approximates $P(W_i=W \mid W_0, \cdots, W_{i-2}, W_{i-1})$ by $f(W_i=W \mid W_{i-2}, W_{i-1})$ where the frequencies f are calculated from a huge "training text" before the recognition task takes place.

One adaptation of the trigram model employs trigrams of POSs to predict the POS of W_i, and frequency of words within each POS to predict W_i itself. Thus, this "3g-gram" model gives $P(W_i=W \mid W_0, \cdots, W_{i-2}, W_{i-1}) \simeq$

$$\sum_{g_j \in G} P(W_i=W \mid g(W_i)=g_j) P(g(W_i)=g_j \mid g(W_{i-2}), g(W_{i-1}))$$

where we let $P(W_i=W \mid g(W_i)=g_j) \simeq f(W_i=W \mid g(W_i)=g_j)$,

$$P(g(W_i)=g_j \mid g(W_{i-2}), g(W_{i-1})) \simeq f(g(W_i)=g_j \mid g(W_{i-2}), g(W_{i-1})).$$

Here G denotes the set of all parts of speech, g_j denotes a particular part of speech, and g (W_i) denotes the part of speech category to which word W_i belongs (abbreviated to g_i from now on); f denotes a frequency calculated from the training text. This "3g-gram" model was used by Derouault and Merialdo for French language modeling [Derouault and Merialdo 1986, 1984], and forms the Markov component of our own model. In practice many POS triplets will never appear in the training text but will appear during the recognition task, so Derouault and Merialdo use a weighted average of triplet and doublet POS frequencies plus a low arbitrary constant to prevent zero estimates for the probability of occurrence of a given POS :

$$P(g_i=g_j \mid g_{i-2}, g_{i-1}) \simeq l_1 * f(g_i=g_j \mid g_{i-2}, g_{i-1}) + l_2 * f(g_i=g_j \mid g_{i-1}) + 10^{-4}.$$

The parameters l_1, l_2 are not constant but can be made to depend on the count of occurrences of the sequence g_{i-2}, g_{i-1} or on the POS of the preceding word, g_{i-1}. In either case these parameters must sum to 0.9999 and can be optimized iteratively; Derouault and Merialdo found that the two weighting methods performed equally well.

The 3g-gram component of our model is almost identical to that of Derouault and Merialdo, although the 153 POSs we use are those of the LOB Corpus. We let l_1 and l_2 depend on the preceding POS g_{i-1}. The cache component keeps track of the **recent** frequencies of words within each POS; it assigns high probabilities to recently used words. Now, let $C_j(W,i)$ denote the cache-based probability of word W at time i for POS g_j . If g $(W) \neq g_j$ then $C_j(W,i)=0$ at all times i, i.e. if W does not belong to POS g_j, its cache-based probability for that POS is always 0. Similarly, let $M_j(W)$ denote the **Markov probability** due to the rest of the pure 3g-gram Markov model. This is approximated by $M_j(W) \simeq f(W_i=W \mid g(W_i)=g_j)$, i.e, the frequency of word W among all words with POS $= g_j$ in the training text.

The final, combined model is then $P(W_i = W) =$

$$\sum_{g_j \in G} P(g_j \mid g(W_{i-2}), g(W_{i-1})) \times [k_{M,j} \times M_j(W) + k_{C,j} \times C_j(W,i)]$$

Here $k_{M,j} + k_{C,j} = 1$; $k_{M,j}$ denotes the weighting given to the "frequency within POS" component and $k_{C,j}$ the weighting of the "cache-based probability" component of POS g_j. One would expect relatively "insensitive" POSs, whose constituent words do not vary much in frequency over time, to have high values of $k_{M,j}$ and low values of $k_{C,j}$; the reverse should be true for "sensitive" POSs. As is described in the next section, approximate values $k_{C,j}$ and $k_{M,j}$ were determined empirically for two POSs g_j to see if these expectations were correct.

These cache-based probabilities $C_j(W,i)$ were calculated as follows. For each POS, a "cache" (just a buffer) with room for 200 words is maintained. Each new word is assigned to a single POS g_j and pushed into the corresponding buffer. As soon as there are 5 words in a cache, it begins to output probabilities which correspond to the relative proportions of words it contains. The lower limit of 5 on the size of the cache before it starts producing probabilities, and the upper size limit of 200, are arbitrary; there are many possible heuristics for producing cache-based probabilities.

3 Implementation and Testing of the Combined Model

3.1 The LOB Corpus

The Lancaster-Oslo/Bergen Corpus of British English consists of 500 samples of about 2000 words each; each word in the corpus is tagged with exactly one of 153 POSs. The samples were extracted from texts published in Britain in 1961, and have been grouped by the LOB researchers into 15 categories spanning a wide range of English prose [Johansson et al 1986, 1982]. We split the tagged LOB Corpus into two unequal parts, one of which served as a training text for our models and the other of which was used to test and compare them. The comprehensiveness of the LOB Corpus made it an ideal training text and a tough test of the robustness of the language model. Furthermore, the fact that it has been tagged by an expert team of grammarians and lexicographers freed us from having to devise our own tagging procedure.

3.2 Parameter Calculation

400 sample texts form the training text used for parameter calculation; the remaining 100 samples form a testing text used for testing and comparison of the pure 3g-gram model with the combined model. Samples were allocated to the training text and the testing text in a manner that ensured that each had similar proportions of samples belonging to the 15 categories identified by the LOB researchers. All parameters for both the pure 3g-gram model and the combined model were calculated from the 400-sample training text.

The two models share a POS prediction component which is estimated by the Derouault-Merialdo method. Triplet and doublet POS frequencies were obtained from 75% (300 of the 400 samples) of the training text; the remaining 25% (100 samples) gave the weights, $l_1(g_{i-1})$ and $l_2(g_{i-1})$, needed for smoothing between these two frequencies. These were computed iteratively using the Forward-Backward algorithm (Derouault and Merialdo [1986], Rabiner and Juang [1986]).

Now the pure 3g-gram model is complete - it remains to find $k_{M,j}$ and $k_{C,j}$ for the combined model. This can be calculated by means of the Forward-Backward method from the 400 samples.

3.3 Testing the Combined Model

As described in 4.2, 80% of the LOB Corpus is used to find the best-fit parameters for a. the pure 3g-gram model b. the combined model, made up of the 3g-gram model plus a cache component. These two models will then be tested on the remaining 20% of the LOB Corpus as follows. Each is given this portion of the LOB Corpus word by word, calculating the probability of each word as it goes along. The probability of this sequence of about 200,000 words as estimated by either model is simply the product of the individual word probabilities as estimated by that model. The higher this overall probability, the better the model. Thus the overall probability is calculated for the pure 3g-gram model and for the combined model; the increase achieved by the latter over the former is the measure of the improvement due to addition of the cache component.

Note that in order to calculate word probabilities, both models must have guessed the POSs of the two preceding words. Thus every word encountered must be assigned a POS. There are three cases :

a). the word did not occur in the tagged training text and therefore is not in the vocabulary;

b). the word was in the training text, and had the same tag wherever it occurred;

c). the word was in the training text, and had more than one tag (e.g. the word "light" might have been tagged as a noun, verb, and adjective).

The heuristics employed to assign tags were as follows :

a). in this case, the two previous POSs are substituted in the Derouault-Merialdo weighted-average formula and the program tries all 153 possible tags to find the one that maximizes the probability given by the formula.

b). in this case, there is no choice; the tag chosen is the unique tag associated with the word in the training text.

c). when the word has two or more possible tags, the tag chosen is the one which makes the largest contribution to the word's probability (i.e. which gives rise to the largest component in the summation on pg. 1).

Thus, although the portion of the LOB Corpus used for testing is tagged, these tags were not employed in the implementation of either model; in both cases the heuristics given above guessed POSs. A separate part of the program compared actual tags with guessed ones in order to collect statistics on the performance of these heuristics.

4 Preliminary Results

1. The first results of our calculations are the values $l_1(g_{i-1})$ and $l_2(g_{i-1})$, obtained iteratively to optimize the weighting between the POS triplet frequency $f(g_i \mid g_{i-2}, g_{i-1})$ and the POS doublet frequency $f(g_i \mid g_{i-1})$ in the estimation of $P(g_i = g_j \mid g_{i-2}, g_{i-1})$. As one might expect, $l_1(g_{i-1})$ tends to be high relative to $l_2(g_{i-1})$ when g_{i-1} occurs often, because the triplet frequency is quite reliable in this case. For instance, the most frequent tag in the LOB Corpus is "NN", singular common noun; we have $l_1(NN) = 0.61$. The tag "HVG", attached only to the word "having", is fairly rare; we have $l_1(HVG) = 0.13$.

However, there are other factors to consider. Derouault and Merialdo state that for g_{i-1} equal to an article, l_1 was relatively low because we need not know the POS g_{i-2} to predict that g_i is a noun or adjective. Thus doublet frequencies alone were quite reliable in this case. On the other hand, when g_{i-1} is a negation, knowing g_{i-2} was very important in making a prediction of g_i, because of French phrases like "il ne veut" and "je ne veux".

Our results from English texts show somewhat different patterns. The tag "AT" for singular articles had an l_1 that was neither high nor low, 0.47 . The tag "CC" for coordinating conjunctions, including "not", had a high l_1 value, 0.80 . Adjectives ("JJ") and adverbs ("RB") had l_1 values even higher than one would expect on the basis of their high frequencies of occurrence : 0.90 and 0.86 respectively.

2. We collected statistics on the success rate of the pure Markov component in guessing the POS of the latest word (using the tag actually assigned the word in the LOB Corpus as the criterion). This rate has a powerful impact on the performance of both models, especially the one with a cache component; each incorrectly guessed POS leads to looking in the wrong cache and thus to a cache-based probability of 0. We are particularly interested in forming an idea of how fast this success rate will increase as we increase the size of the training text.

Of the words that had occurred at least once in the training text, 83.9 % had tags that were guessed correctly (16.1 % incorrectly). Words that never occurred in the training text were assigned the correct tag only 22 % of the time (78 % incorrect). Apparently the information contained in the counts of POS triplets, doublets, and singlets is a good POS predictor when combined with some knowledge of the possible tags a word may have, but not nearly as good on its own.

Among the words that appeared at least once in the training text, a surprisingly high proportion - 42.8 % - had more than one possible POS. Of these, 66.7 % had POSs that were guessed correctly. Thus it might appear that performance is degraded when the program must make a choice between possible tags. This analysis is faulty; a given word might have

many POSs, and perhaps the correct one was not found in the training text at all. The most important statistic , therefore, is the proportion of words in the testing text whose tag was guessed correctly among the words that had also appeared with the correct tag in the training text. This proportion is 94.0 %. It seems reasonable to regard this as being an indication of the upper limit for the success rate of POS prediction with training texts of manageable size; it provides an estimate of the success rate when the two main sources of error (words found in the testing text but not the training text, words found in both texts which are tagged in the testing text with a POS not attached to them anywhere in the training text) are eliminated.

3. We have not yet tested the full combined model (with a cache component and a Markov component) against the 3g-gram Markov model. However, we have examined the effect on the predictive power of the Markov model of including cache components for two POSs : singular common noun (label "NN" in the LOB Corpus) and preposition (label "IN" in the LOB Corpus). These two were chosen because they occur with high frequency in the Corpus, in which there are 148,759 occurrences of "NN" and 123,440 occurrences of "IN", and because "NN" is a content word category and "IN" a function word category. Thus they provide a means of testing the hypothesis outlined in the Introduction, that a cache component will increase predictive power for content POSs but not make much difference for function POSs.

For both POSs, the expectation that the 200-word cache will often contain the current word was abundantly fulfilled. On average, if the current word was an NN-word, it was stored in the NN cache 25.8 % of the time; if it was an IN-word, it was stored in the IN cache 64.7 % of the time. The latter is no surprise - there are relatively few different prepositions - but the former figure is remarkably high, given the large number of different nouns. Note that the figure would be higher if we counted plurals as variants of the singular word (as we may do in future implementations).

We have not yet obtained the best-fit weighting for the combined model. However, we tried 3 different combinations for the NN-words and the IN-words. If "a" is the weight for the cache component and "b" the weight for the Markov component, the 3 combinations (a, b) are (0.2, 0.8), (0.5, 0.5), and (0.9, 0.1); the pure Markov model corresponds to the weighting (0.0, 1.0). To assess the performance of each combination for NN-words and IN-words, we calculated i). the log product of the estimated probabilities for NN-words only under each of the 4 formulas ii). the log product of the estimated probabilities for IN- words only under each of the 4 formulas. It is then straightforward to calculate the improvement per word obtained by using a cache instead of the pure Markov model.

For NN-words, the (0.2, 0.8) weighting yielded an average multiple of 2.3 in the estimated probability of a word in the testing text over the probability as calculated by the pure Markov model ; the (0.5, 0.5) weighting yielded a multiple of 2.0 per word, and the (0.9, 0.1) actually decreased the probability by a factor of 1.5 per word.

For IN-words, the (0.2, 0.8) weighting gave an average multiple of 5.1, the (0.5, 0.5) weighting a multiple of 7.5 and the (0.9, 0.1) weighting a multiple of 6.2 .

5 Conclusions

The preliminary results listed above seem to confirm our hypothesis that recently-used words have a higher probability of occurrence than the 3g-gram model would predict. Surprisingly, if the above comparison of the POS categories "NN" and "IN" is a reliable guide, this increased probability is more dramatic in the case of content-word categories. Perhaps the smaller number of different prepositions makes the cache-based probabilities more reliable in this case.

Since the cost of maintaining a 200-word cache, in terms of memory and time, is modest, and the increase in predictive power can be great, the approach outlined above should be considered as a simple way of improving on the performance of a 3g-gram language model for speech recognition. If memory is limited, one would be wise to create caches only for POSs that occur with high frequency and ignore other POSs.

Our immediate goal is to build caches for a larger number of POSs, and to obtain the best-fit weighting for each of them, in order to test the full power of the combined model. Eventually, we may explore the possibility of ignoring variations in the exact form of a word, merging the singular form of a noun with its plural, and different tenses and persons of a verb.

This line of research has more general implications. The results above seem to suggest that at a given time, a human being works with only a small fraction of his vocabulary. Perhaps if we followed an individual's written or spoken use of language through the course of a day, it would consist largely of time spent in language "islands" or sublanguages, with brief periods of time during which he is in transition between islands. One might attempt to chart these "islands" by identifying groups of words which often occur together in the language. If this work is ever carried out on a large scale, it could lead to pseudo-semantic language models for speech recognition, since the occurrence of several words characteristic of an "island" makes the appearance of **all** words in that island more probable.

Bibliography

1. R. Campo, L. Fissore, A. Martelli, G. Micca, and G. Volpi, "Probabilistic Models of the Italian Language for Speech Recognition". *Recent Advances and Applications of Speech Recognition* (international workshop), pp. 49-56, Rome, May 1986.

2. A.M. Derouault and B. Mérialdo., "Natural Language Modeling for Phoneme-to-Text Transcription", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, pp. 742-749, No. 1986.

3. A.M. Derouault and B. Mérialdo, "Language Modeling at the Syntactic Level", *7th Int. Conf. Pattern Recognition*, Vol. II, pp. 1373-1375, Montreal, Aug. 1984.

4. W.N. Francis, "A Tagged Corpus - Problems and Prospects", in *Studies in English Linguistics for Randolph Quirk*, S. Greenbaum, G. Leech, and J. Svartvik, Eds. London: Longman, 1980, pp. 193-209.

5. F. Jelinek, "The Development of an Experimental Discrete Dictation Recognizer", *Proc. IEEE*, Vol. 73, No.11, pp 1616-1624, Nov. 1985.

6. F. Jelinek, R.L. Mercer, and L.R. Bahl, "A Maximum Likehood Approach to Continuous Speech Recognition", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 179-90, Mar. 1983.

7. F. Jelinek, "Markov Source Modeling of Text Generation", personal communication.

8. F. Jelinek, "Self-Organized Language Modeling for Speech Recognition", personal communication.

9. S. Johansson, E. Atwell, R. Garside, and G. Leech, *The Tagged LOB Corpus Users Manual.* Norwegian Computing Centre for the Humanities, Bergen, 1986.

10. S. Johansson, ed, *Computer Corpora in English Language Research.* Norwegian Computing Centre for the Humanities, Bergen 1982.

11. S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "An Introduction to the Application of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", *The Bell System Technical Journal*, Vol. 62, No. 4, pp. 1035-1074, Apr. 1983.

12. I. Marshall, "Choice of Grammatical Word-Class Without Global Syntactic Analysis: Tagging Words in the LOB Corpus", *Computers and the Humanities*, Vol. 17, No. 3, pp. 139-150, Sept. 1983.

13. A. Martelli, "Probability Estimation of Unseen Events for Language Modeling", personal communication.

14. E.M. Muckstein, "A Natural Language Parser with Statistical Applications", *IBM Research Report* RC7516 (#38450), Mar. 1981.

15. A. Nadas, "Estimation of Probabilities in the Language Model of the IBM Speech Recognition System", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 32, pp. 859-861, Aug. 1984.