Variable-Length Sequence Modeling: Multigrams

Frédéric Bimbot, Roberto Pieraccini, Esther Levin, and Bishnu Atal

***Abstract*— The conventional n-gram language model exploits dependencies between words and their fixed-length past. This letter presents a model that represents sentences as a concatenation of variable-length sequences of units and describes an algorithm for unsupervised estimation of the model parameters. The approach is illustrated for the segmentation of sequences of letters into subword-like units. It is evaluated as a language model on a corpus of transcribed spoken sentences. Multigrams can provide a significantly lower test set perplexity than n-gram models.**

I. Introduction

SPOKEN and written language results from a complex encoding of a message into a stream of acoustic events or symbolic units. This process is subject to several constraints: Sequences of phones must be pronounceable, combinations of phonemes or letters must form words, and grammar rules govern word order. These constraints induce a certain degree of redundancy in the encoding process, a consequence of which is the existence of typical sequences in the stream of units observed in natural language. For instance, typical sequences of letters in written texts are the expression of its lexical structure, and typical sequences of words are the consequence of the use of syntactic rules. Classical probabilistic models, such as n-grams, exploit this redundancy using a fixed-length context.

In this letter, we propose a variable-length sequence model: the multigram model. According to our approach, a stream of symbolic units is assumed to be generated as a succession of variable-length independent sequences. We first present a mathematical formulation of the model. Then, we describe an algorithm for estimating its parameters. We illustrate its results on the segmentation of streams of letters, and we evaluate its performance as a language model on a corpus of spoken sentences known as ATIS [2].

II. Formulation

The classical n-gram model, as presented, for instance, by Jelinek [1], estimates the likelihood of a string $\boldsymbol{W}$ of N units $w_1 w_2 \cdots w_N$ as

$$\mathcal{L}(W) = \mathcal{L}(w_1\, w_2\, \cdots\, w_N) = \prod_{i=1}^{N} P(w_i / w_{i-n+1}\, \cdots\, w_{i-1}).$$

The underlying hypothesis of this approach is that the probability of a given word depends only on the $n-1$ previous words. The considered history has, therefore, a fixed length, which is obviously too simplistic a constraint for some natural language tasks.

In opposition, the multigram model estimates the likelihood of a string $\boldsymbol{W}$ as the likelihood of the best segmentation of $\boldsymbol{W}$ into variable-length sequences. These sequences are supposed to be independent from one another.

A segmentation $\boldsymbol{S}$ of string $\boldsymbol{W}$ into Q_S sequences can be written $\boldsymbol{S} = s_1 s_2 \cdots s_k \cdots s_{Q_S}$. Each sequence s_k corresponds to a substring of $\boldsymbol{W}$ starting at index i_k and finishing at index $i_{(k+1)} - 1$, i.e., $s_k = [w_{i_k} w_{i_k+1} \cdots w_{i_{(k+1)}-1}]$. A segmentation of $\boldsymbol{W}$ into Q_S sequences is therefore described by a set $\boldsymbol{B_S} = \{i_1, i_2, \cdots, i_{Q_S}, i_{Q_S+1}\}$ of indexes marking the first word of each sequence with the following constraints: $i_1 = 1, i_{Q_S+1} = N+1$ and $i_1 < i_2 < \cdots < i_{Q_S+1}$. If we restrict to n the maximum length of any sequence, we have the extra constraint: For $1 \le k \le Q_S, 1 \le i_{k+1} - i_k \le n$. If we finally denote as $\{\boldsymbol{B_S}\}$ the set of all possible sets $\boldsymbol{B_S}$ satisfying the constraints above, and if we assume that sequences are statistically independent, the n-multigram model expresses the likelihood of the string $\boldsymbol{W}$ as

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}) &= \max_{\{\boldsymbol{S}\}} \prod_{k=1}^{Q_S} P(s_k)\\ &= \max_{\{\boldsymbol{B_S}\}} \prod_{k=1}^{Q_S} P([w_{i_k} w_{i_k+1} \cdots w_{i_{(k+1)}-1}]).\end{aligned}$$

For instance, for a string of four words and a 3-multigram model, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}) &= \mathcal{L}(w_1\, w_2\, w_3\, w_4)\\ &= \max\{P([w_1\, w_2\, w_3])P([w_4]),\\ &\quad P([w_1])P([w_2\, w_3\, w_4]),\\ &\quad P([w_1\, w_2])P([w_3\, w_4]),\\ &\quad P([w_1\, w_2])P([w_3])P([w_4]),\\ &\quad P([w_1])P([w_2\, w_3])P([w_4]),\\ &\quad P([w_1])P([w_2])P([w_3 w_4]),\\ &\quad P([w_1])P([w_2])P([w_3])P([w_4])\}.\end{aligned}$$

III. Algorithm

The estimation of the model on a training corpus $\boldsymbol{W}$ can be decomposed in three procedures: An initial estimation of the sequence probabilities (Section III-A), a segmentation (Section III-B), and a reestimation of the probabilities (Section III-C). The steps in Sections III-B and C are repeated iteratively.

Manuscript received December 13, 1994; approved March 15, 1995. The associate editor coordinating the review of this letter and approving it for publication was Dr. L. Niles. This work was carried out by while F. Bimbot was visiting AT&T Bell Laboratories.

F. Bimbot is with ENST, Dept. Signal, CNRS, Paris, France.

R. Pieraccini, E. Levin, and B. Atal are with the Speech Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974 USA.

IEEE Log Number 9412254.

A. Initial Estimation of Sequence Probabilities:

Let $c_{\text{init}}(s)$ denote the number of occurrences of a given substring s within the training corpus $\boldsymbol{W}$ of length N. Let C_k be the number of substrings of length $k \leq n$. With

$$C_{init} = \sum_{k=1}^{n} C_k \approx nN$$

an initial estimate of the probability of sequence s is obtained as

$$P_{init}(s) = \frac{c_{init}(s)}{C_{init}}.$$

B. Segmentation

If we suppose that the probabilities of all sequences are known, a given string $\boldsymbol{W}$ can be parsed according to the maximum likelihood segmentation by a Viterbi procedure. For the partial sequence of $\boldsymbol{W}$ up to index $k+1$, the likelihood $\mathcal{L}(w_1 \cdots w_{k+1})$ is computed as

$$\max_{1 \leq i \leq n} \{P([w_{k-i+2} \cdots w_{k+1}])\mathcal{L}(w_1 \cdots w_{k-i+1})\}.$$

This formula is used recursively for parsing. The best parsing of the whole string $\boldsymbol{W}$ provides a maximum likelihood segmentation.

C. Reestimation

After segmentation, the training set is parsed into C separate sequences of maximal length n. If we now denote as $c(s)$ the number of sequences s in the segmented corpus, we can reestimate the probability of sequence s as

$$P(s) = \frac{c(s)}{C}.$$

The main difference with Section III-A is that the initial estimation uses all co-occurrences of symbols up to length n within the input set, whereas the reestimation step only considers entire sequences resulting from the parsing of the previous iteration. With these new estimates, the training corpus can be parsed again (Section III-B), and the process is iterated until convergence is reached or after a fixed number of iterations. All sequences that are obtained constitute a dictionary of multigrams.

D. Pruning

In order to improve generalization on a test corpus, we found it useful to penalize sequences having low frequencies. A possible way of doing this is to replace the probability estimate used above by the lower bound of the confidence interval of this estimate, i.e., we replace $P(s)$ by

$$P'(s) = \frac{c(s)}{C}\left(1 - a\sqrt{\frac{C - c(s)}{C.c(s)}}\right).$$

With a pruning factor $a = 1.96, P'$ is the lower bound of the 95% confidence interval. When $P' < 0, P'$ is set to 0. With such an estimate, the sum of "probabilities" gets lower than 1, which requires a renormalization. P' can be interpreted as a "worst case" estimate.

Input text :

```
1 blessedisthemanthatwalkethnotinthecounseloftheungodlynorsta
  ndethinthewayofsinnersnorsittethintheseatofthescornful
2 buthisdelightisinthelawofthelordandinhislawdothhemeditateda
  yandnight
3 andheshallbeliketatreeplantedbytheriversofwaterthatbringethf
  orthhisfruitinhisseasonhisleafalsoshallnotwitherandwhatsoev
  erhedoethshallprosper
4 theungodlyarenotsobutarelikethechaffwhichthewinddrivethaway
5 thereforetheungodlyshallnotstandinthejudgmentnorsinnersinth
  econgregationoftherighteous
```

Output text :

```
1 bless ed isthe man that walk eth not inthe couns el ofthe
  un godly n or stand eth inthe way of sin n ers n or sitte
  th inthe se at ofthe s corn ful
2 but his d e light is inthe law ofthe lord and inhis law do
  th he medit at e day and night
3 and he shall be likea t re e plant ed bythe river sof water
  that bring eth forth his fruit inhis s eason his l ea f
  also shall not with erand what so ever he doeth shall prosp
  er
4 the un godly are not so but are like the ch a f f which
  thew in d drive th away
5 ther efore the un godly shall not stand inthe judgm ent n
  or sin n ers inthe co ngreg ation ofthe right eous
```

Fig. 1. Old Testament (Psalms)—King James version. Excerpts of the input and output texts. (First five verses.) 5-multigram model with pruning factor $a = 2.0$. Borders between output sequences are marked by spaces.

IV. Illustration

Fig. 1 shows the result of the multigram sequence modeling technique on an English text from which all spaces between words were removed. Our corpus—the Psalms of the Old Testament (King James version)—contains approximately 200 000 characters. The set of elementary units, in this case, consists of the 26 letters of the alphabet. No distinction was made between uppercase and lowercase letters, and punctuation symbols were removed. In this section, the multigram model is implemented in order to retrieve typical sequences of letters in strings of concatenated words.

For this experiment, the probabilities were estimated as described above, for a 5-multigram model, with a pruning factor $a = 2.0$. After 10 iterations, convergence was observed, and the dictionary contained approximately 1100 multigrams. Some typical English words or morphemes are extracted. Some frequent combinations of small words are merged together (*inthe, ofthe, inhis,* ...). Rare words are broken into smaller units (*se at, l ea f, ch a f f,* ...). Occasionally, an unappropriate segmentation occurs such as *river sof* for "rivers of," *thew ind* for "the wind," ...

V. Evaluation

We also tested the method for language modeling. As opposed to the previous illustration on letters, the elementary unit in this section is the word. The goal of the algorithm is therefore to extract typical sequences of words. We compare the multigram model to classical n-gram models.

A. Perplexity

The objective measure of performance is the test set perplexity from the test set $\boldsymbol{W}_{\text{test}}$ of size N_{test}:

$$X = 2^H \quad \text{where} \quad H = -\frac{1}{N_{\text{test}}} \log_2[\mathcal{L}(\boldsymbol{W}_{\text{test}})].$$

TABLE I
PERPLEXITY OF n-GRAM MODELS ON THE ATIS DATABASE

n-gram order	*n* = 1	*n* = 2	*n* = 3	*n* = 4	*n* = 5
nb of *n*-grams	931	7253	16589	24121	27979
training set perpx	80.5	11.8	6.5	5.2	4.7
test set perpx	91.5	21.0	33.1	69.5	123.4

TABLE II
PERPLEXITY OF 5-MULTIGRAM MODELS (ATIS DATABASE) WITH SEVERAL PRUNING FACTORS ($0.0 \leq a \leq 5.0$)

5-multigrams	*a* = 0.0	*a* = 1.0	*a* = 2.0	*a* = 3.0	*a* = 5.0
nb of sequences	8928	4328	2256	1569	1218
training set perpx	6.9	9.3	13.0	15.6	19.3
test set perpx	27.6	18.4	17.7	20.5	26.0

The perplexity X indicates the degree of difficulty to predict the language by the probabilistic model.

B. Unseen Units

An arbitrary nonzero probability is given to all sequences of length equal to 1 that do not exist in the dictionary of multigrams:

$$\text{if } P([w_t]) = 0,$$
$$\text{we set: } P([w_t]) = P_o = \frac{1}{2N},$$

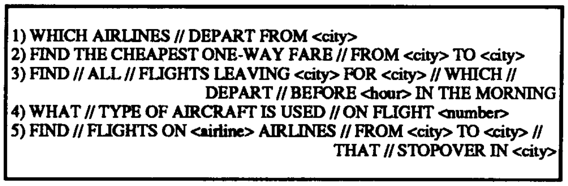
where N is the number of words in the training corpus. The consequence of this setting of the minimum probability is that any test string can at worst be parsed into individual symbols.

C. n-Gram Model

All n-gram probabilities are computed as their frequency in the training corpus. For unseen n-grams in the test set, we assign a constant penalty of P_o, as defined above. This method, though nonoptimal, is consistent with the approach we adopted for unseen multigrams.

D. Corpus

The training corpus is the text of the spontaneous utterances of the ATIS database [2]. It contains more than 10 000 sentences and more than 100 000 words. City names, month names, day names, airline names, hours, and numbers were each replaced by a specific symbol. The vocabulary has about 900 words. The test corpus is another set of 1000 such sentences (≈10 000 words). The performance of n-grams ($1 \leq n \leq 5$) are compared to the 5-multigrams, with several pruning factors ($0.0 \leq a \leq 5.0$). Tables I and II summarize the results in terms of number of sequences, training, and test set perplexity.



```
1) WHICH AIRLINES // DEPART FROM <city>
2) FIND THE CHEAPEST ONE-WAY FARE // FROM <city> TO <city>
3) FIND // ALL // FLIGHTS LEAVING <city> FOR <city> // WHICH //
                               DEPART // BEFORE <hour> IN THE MORNING
4) WHAT // TYPE OF AIRCRAFT IS USED // ON FLIGHT <number>
5) FIND // FLIGHTS ON <airline> AIRLINES // FROM <city> TO <city> //
                                             THAT // STOPOVER IN <city>
```

Fig. 2. ATIS database-test set. Excerpt of the output text. 5-multigram model with pruning factor $a = 2.0$. Borders between sequences are marked by a double slash.

E. Results

With a number of sequences two to four times smaller than the number of bigrams, the 5-multigram approach, with a pruning factor between 1.0 and 3.0, provides a significantly lower test set perplexity. It can also be noted that 3-grams, 4-grams and 5-grams are more efficient than 5-multigrams for modeling the training corpus but that 5-multigrams generalize better on the test set. Fig. 2 shows an example of the segmentation obtained on test sentences with a 5-multigram model and a pruning factor $a = 2.0$. As can be seen from this example, groups of words extracted by the model show interesting relations with syntactic groups and semantic concepts. The model may prove useful in natural language understanding.

VI. CONCLUSIONS

The n-multigram sequence modeling technique makes a different assumption compared with the classical n-gram approach: A string of units is decomposed into variable-length sequences that are supposed to be independent from each other. The structuring of a training corpus into a succession of such variable-length sequences is obtained through a totally unsupervised process. The new model seems to be an interesting challenger for the n-gram model, for the natural language task that we have experimented. We believe that the multigram model deserves a more systematical investigation.

REFERENCES

[1] F. Jelinek, "Self-organized language modeling for speech recognition," in *Readings in Speech Recognition* (A. Waibel and K. F. Lee, Eds.). San Mateo, CA: Morgan Kaufmann, 1990, pp. 450–506.

[2] MADCOW, "Multidata collection for a spoken language corpus," in *Proc. 5th DARPA Workshop Speech and Natural Language*. Harriman, NY, 1992, pp. 7–14.