STATISTICAL LANGUAGE MODELING WITH A CLASS BASED
N-MULTIGRAM MODEL

Sabine Deligne

IBM T. J. Watson Research Center P. O. Box 218,
Yorktown Heights, NY 10598
deligne@us.ibm.com

ABSTRACT
In this paper, we report on speech recognition experiments with an \( n \)-multigram language model, a stochastic model which assumes dependencies of length \( n \) between variable-length phrases. The \( n \)-multigram probabilities can be estimated in a class-based framework, where both the phrase distribution and the phrase classes are learned from the data according to a Maximum Likelihood criterion, using a generalized Expectation-Maximization algorithm. In our speech recognition experiments on a database of air travel reservations, the 2-multigram model allows a reduction of 10% of the word error rate with respect to the usual trigram model, with 25% fewer parameters than in the trigram model. We also report on a scheme where some a priori information is introduced in the model via semantic tagging.

1. INTRODUCTION
In this paper, we report on speech recognition experiments with an \( n \)-multigram language model, a stochastic model which assumes dependencies of length \( n \) between variable-length phrases. More specifically, we address the issue of estimating \( n \)-multigram probabilities in a class-based framework, where both the word dependencies and the equivalence classes are learned from the data. The multigram approach was introduced in [1], and in [2] it was used to derive variable-length phrases under the assumption of independence of the phrases. Various ways of theoretically relaxing this assumption are given in [3]. Experiments with 2-word multigrams embedded in a deterministic variable \( n \)-gram scheme are reported in [4]. Section 2 of this paper describes the class based \( n \)-multigram model and its training procedure. In section 3, we present speech recognition experiments with \( n \)-multigrams and \( n \)-grams on a database of airline reservations. In section 4, we report on experiments that integrate semantic tagging with the \( n \)-multigram training procedure. Conclusions and future directions for this work are given in section 5.

2. THE \( N \)-MULTIGRAM MODEL
In the multigram framework, the sentences are structured into variable-length phrases called multigrams. The likelihood of a sentence is computed by summing up the likelihoods of all the segmentations of the sentence into phrases. The likelihood of any particular segmentation into phrases is computed by assuming that each phrase depends on the \((n - 1)\) previous phrases, hence the name \( n \)-multigram model. Let \( W \) denote a string of words, \( l \) the maximum length of a phrase, and \( \{ S \} \) the set of possible segmentations of \( W \). The likelihood of \( W \) is:

\[
\mathcal{L}(W) = \sum_{S \in \{S\}} \mathcal{L}(W, S) \tag{1}
\]

and, assuming that each phrase depends on the \((n - 1)\) phrases, the likelihood of a segmentation \( S \) of \( W \) is:

\[
\mathcal{L}(W, S) = \prod_{\tau} p(s_{\tau - n + 1} \ldots s_{\tau - 1} | s_{\tau}) \tag{2}
\]

with \( s_{\tau} \) denoting the \((\tau)\)th phrase in the segmentation \( S \). The model is thus fully defined by the set of \( n \)-gram probabilities on the set \( \{s_i\} \), of all the phrases which can be formed by combining 1, 2, ... up to \( l \) words of the vocabulary. Throughout this paper, we refer to such a set of probabilities as an \( n \)-multigram distribution. In the class version of the model, the set of phrases is partitioned into equivalence classes. Denoting by \( q \) a class membership function, which specifies for each sequence \( s_i \) the class \( \mathcal{C}_q(s_i) \) it belongs to, and assuming that each class depends on the \((n - 1)\) previous classes, the likelihood \( \mathcal{L}(W, S) \) in the class version of the model is:

\[
\prod_{\tau} p(\mathcal{C}_q(s_{\tau}) | \mathcal{C}_q(s_{\tau - n + 1}) \ldots \mathcal{C}_q(s_{\tau - 1})) p(s_{\tau} | \mathcal{C}_q(s_{\tau})) \tag{3}
\]

Be it in equation (2) or (3), the likelihood of a corpora is a function of the \( n \)-multigram distribution and the class membership function \( q \):

\[
\mathcal{L}(W) = F(\{ p(s_{i_n} | s_{i_1} \ldots s_{i_{n-1}} \})_{i_1 \ldots i_n}, q) \tag{4}
\]

The problem of estimating an \( n \)-multigram model in an ML fashion thus consists in estimating an \( n \)-multigram distribution and in defining a class membership function maximizing equation (4). For lack of a closed form solution, we define an iterative procedure where equation (4)
is alternately optimized, with respect to the \( n \)-multigram distribution, and with respect to the class membership function. Assuming that \( p^{(k)} \) and \( q^{(k)} \), the \( n \)-multigram distribution and the class membership function at iteration \( (k) \) are known, then iteration \((k+1) \) consists of:

**Step (1): estimation of a new \( n \)-multigram distribution \( p^{(k+1)} \), such that:**

\[
F(p^{(k+1)} , q^{(k)}) \geq F(p^{(k)} , q^{(k)})
\]

**Step (2): definition of a new class membership function \( q^{(k+1)} \), such that:**

\[
F(p^{(k+1)} , q^{(k+1)}) \geq F(p^{(k+1)} , q^{(k)})
\]

The estimation of an \( n \)-multigram distribution (step (1) of each training iteration) can be addressed as an ML estimation problem from missing data [5]. The basic idea behind this approach is that the statistics which are missing to compute the ML estimates of parameters can be approximated by their expected values. In [6], we show that the EM equation to estimate an \( n \)-multigram distribution is:

\[
p_{(k+1)}(s_{i_n} | s_{i_1} \ldots s_{i_{n-1}}) = \\
\frac{\sum_{\{S\}} c(s_{i_1} \ldots s_{i_n} | S) L^{(k)}(S | W)}{\sum_{\{S\}} c(s_{i_1} \ldots s_{i_{n-1}} | S) L^{(k)}(S | W)}
\]

(5)

In practice, equation (5) is implemented with a forward-backward algorithm so that the complexity of the algorithm is \( O(l N T) \), with \( l \) the maximum size of a phrase and \( T \) the number of words in the corpus. The forward-backward equations are given in [6].

The optimization of the class membership function (step (2) of each training iteration) aims at partitioning the set of phrases in a way maximizing the data likelihood, given a known \( n \)-multigram distribution and given a prespecified number of classes. In our experiments, we use the clustering algorithm presented in [7]: starting from any given classification, each iteration consists in removing each phrase from its current class and in assigning it to the class for which the data likelihood is maximal. The clustering iterations are applied till either the number of class exchanges becomes zero, or after a prespecified number of iterations.

The \( n \)-multigram distribution estimated in the class version of the model can be used to smooth an \( n \)-multigram distribution obtained without phrase clustering. ML estimates of linear interpolation weights can be estimated on some held-out data using an EM procedure [6]. In addition to the estimation and classification steps, the algorithm includes some precautions in order to limit the effect of overtraining:

- the combinations of words occurring less than \( thr1 \) times are not registered at the initialization, and the \( n \)-uplets of phrases involving a phrase occurring less than \( thr2 \) times\(^1\)

\(^1\)The threshold values \( thr1 \) and \( thr2 \) are set empirically depending on the degree of sparseness of the training data; in our experiments we observed that setting \( thr2 \) to about half the value of \( thr1 \) yielded the best performances.

### 3. SPEECH RECOGNITION EXPERIMENTS

All experiments are conducted on an Air Travel Information System database (ATIS) collected over the phone. It consists of utterances by customers asking for air travel reservations. The partition of the data into training, held-out and test data is described in Table 1. The parameters of both the word and class versions of the models are estimated on the training set. The held-out data are used to estimate interpolation weights between word based and class based models. All perplexity values and recognition error rates are computed on the test set. The test set consists of calls from more than 30 different speakers. Experiments are reported for phrases having at most

<table>
<thead>
<tr>
<th>Nb sentences</th>
<th>Train</th>
<th>Held-out</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>81,463</td>
<td>20,000</td>
<td>689</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nb tokens</th>
<th>709,940</th>
<th>146,963</th>
<th>2,745</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>+136 OOV</th>
<th>+13 OOV</th>
</tr>
</thead>
</table>

Table 1: ATIS database used in all the experiments.

\( l = 1, 2, 3, 4 \) or 5 words (for \( l = 1 \), bi-multigrams correspond to conventional bigrams). The dictionaries of phrases are pruned by discarding all the phrases occurring less than \( thr1 = 200 \) times at initialization, and less than \( thr2 = 100 \) times after each iteration, except for the one-word phrases which are kept with a number of occurrences set to one. The bi-multigram probabilities are estimated with 3 iterations of equation (5). In the class based models, the phrases are clustered into 100 phrase-classes using the exchange algorithm. The phrases occurring less than 3 times are not clustered, instead they are assigned to a common class. The \( n \)-gram models are trained with the CMU-Cambridge toolkit [8]. Both the bi-multigram and the \( n \)-gram estimates are smoothed with the backoff smoothing technique [9] using Witten-Bell discounting [10].

The baseline recognition lexicon contains the phonetic baseforms of all the words occurring in the training, held-out and test sets. When decoding with a bi-multigram model, the phrases having a non zero probability are added as compound words to this baseline lexicon with all the possible combinations of the word pronunciation variants. The speech recognition system used in these experiments is a speaker independent recognizer based on IBM technology and trained on a few hundred hours of either phone or band-limited speech data. Only a tiny portion of the speech data used to train the acoustic models is related to the ATIS task. The system is an HMM-based recognizer with 39,133 gaussians modeling 2,615 context-dependent subphone units. The front end of the system...
computes feature vectors comprising 12 cepstra plus the energy, plus delta and delta-delta coefficients computed from 10 ms frames.

In Tables 2, 3 and 4, we report on recognition experiments with, respectively, bigram and trigram models, bimultigram models, and models resulting from the interpolation of bimultigram and class based bimultigram models. The bimultigram models yield recognition scores very similar to the trigram scores, while being smaller models, which illustrates their ability to select the most relevant phrases. For example, the word error rate with a bimultigram model allowing up to 5-word long phrases is slightly lower than the trigram word error rate (17.6% versus 17.2%), and besides it has 35% fewer parameters (57,618 versus 89,696). When the bimultigram models are interpolated with their class counterpart, the word error rate further decreases. The word error rate with the interpolated bimultigram model with 5-word long phrases drops to 15.6%, i.e. about 10% less than the trigram word error rate. Besides the interpolated model is still significantly smaller than the trigram model: about 25% fewer parameters (68,675 versus 89,696).

<table>
<thead>
<tr>
<th>n-gram model</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model size</td>
<td>33,151</td>
<td>89,696</td>
</tr>
<tr>
<td>WER (%)</td>
<td>18.9</td>
<td>17.2</td>
</tr>
</tbody>
</table>

Table 2: Model size and word error rate with 2-gram and 3-gram models.

<table>
<thead>
<tr>
<th>2-multigram model</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model size</td>
<td>33,151</td>
<td>59,170</td>
<td>56,468</td>
<td>55,759</td>
<td>57,618</td>
</tr>
<tr>
<td>WER (%)</td>
<td>18.9</td>
<td>17.0</td>
<td>17.1</td>
<td>17.3</td>
<td>17.0</td>
</tr>
</tbody>
</table>

Table 3: Model size and word error rate with 2-multigram models (F.-B. training).

<table>
<thead>
<tr>
<th>Interpolated 2-multigram model</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model size</td>
<td>42,853</td>
<td>69,664</td>
<td>67,219</td>
<td>66,652</td>
<td>68,675</td>
</tr>
<tr>
<td>WER (%)</td>
<td>18.4</td>
<td>16.9</td>
<td>16.3</td>
<td>17.0</td>
<td>15.6</td>
</tr>
</tbody>
</table>

Table 4: Model size and word error rate when interpolating non-class and class based 2-multigram models (100 classes, F.-B. training).

4. USE OF SEMANTIC TAGS

In [6], we noticed how intra-class phrases often display a strong semantic correlation, even though the clustered phrases in these experiments were derived fully blindly, i.e. with no a priori information at all. Here, we investigate the introduction of a priori information via semantic tagging to enable the class based model to generalize to unseen phrases. We define 12 semantic tags matching the ATIS domain (@AIRLINES, @AIRPORTNAME, @CITYNAME, @COUNTRYNAME, @DAYNAME, @DAYNUMBER, @FLIGHTNUMBER, @HOUR, @MINUTE, @MONTHNAME, @STATENAME, @YEARNUMBER) and we use them to label the training data. The labelling is performed with INTEX [11], a linguistic development tool2 that allows users to build finite state descriptions of natural language and to apply them to large corpora. In our experiments, we build a finite state transducer to describe the sequences of words covered by each semantic tag. Each transducer was set to output its semantic tag, so that the training data were automatically labelled by applying the set of transducers.

During the training of the language model, the sequences of words bearing a label were replaced by their label. In the class based model, it results in a two-level classification scheme, since the semantic labels can be involved in phrases which are in turn clustered. In Table 5, we show some examples of classes of phrases that were obtained by training a 2-multigram model with phrases of up to 4 words on the labelled training data. Once the language model is trained, each phrase which contains semantic tags has its probability redistributed across all the phrases that can be derived by expanding the tags: the phrase BETWEEN+@CITYNAME+AND+@CITYNAME expands into: BETWEEN+ATLANTA+AND+DALLAS...etc. Even if the phrase BETWEEN+ATLANTA+AND+BOSTON is not observed in the training corpus, its probability can be computed by the model. We experimented this scheme to rescore lattices output by a speech recognition system. Rescoring the lattices with n-multigram models trained on the labelled data allowed us to reduce by about 20% the word error rate with respect to rescoring with n-multigram models trained on the plain words. However, it did not outperform the results showed in Table 4, where the multigram models are applied by the recognition stage. In our experiments, each tag was expanded into all the sequences of words that were labelled with it in the data, and the probability of the phrases with tags was equally redistributed across all possible expansions. However, this scheme could be taken advantage of by accommodating the expansion of the tags so as to fit a specific application within the ATIS domain: for example the list of city names could be restrained to contain only specific cities, and the probability of each city within the tag could be weighted based on the knowledge of the application.

5. CONCLUSIONS AND PERSPECTIVES

In this paper, we presented a stochastic class phrase based model for which we proposed a training algorithm based

---
2A complete description of the INTEX project can be found at www.ladl.inria.fr.
constraints its pronunciation. Besides, it would be interesting to see how learning trigram dependencies between phrases would compare to this, though it might require to accommodate the training procedure in order to manage with the complexity of the algorithm which grows exponentially with n.

6. REFERENCES


