A LANGUAGE MODEL COMBINING N-GRAMS AND STOCHASTIC FINITE STATE AUTOMATA*

Alexis Nasr, Yannick Estève, Frédéric Béchet, Thierry Spriet, Renato de Mori
LIA - University of Avignon
BP 1228 - 84911 Avignon Cedex 9 - France
alexis.nasr@lia.univ-avignon.fr

1 ABSTRACT

This paper describes a new kind of language models composed of several local models and a general model linking the local models together. Local models describe more finely subparts of the textual data than a conventional n-gram trained on the complete corpus. They are built on lexical and syntactic criteria. Both local and global models are integrated in a single hidden Markov model. Experiments showed a 14% decrease in perplexity compared to a bigram model on a small corpus of telephonic communications.

2 INTRODUCTION

Language models described in this paper are based on the intuitive idea that different parts of sentences in a given corpus could be described more finely by local models rather than by a conventional n-gram model trained on the entire corpus, especially when not enough data is provided to train a trigram. We propose to implement this idea by bringing together some word sequences appearing in a corpus. Every grouping will constitute a subcorpus, we will call a class, on which a local model is trained. The structure of the word sequences gathered together in a class corresponds to well formed syntactic units or phrases, as noun phrases, prepositional phrases or verb phrases, of variable length. Phrases are gathered together according to the similarity of their distribution in a training corpus. Two advantages are expected from this kind of models. The first is better reaction to data sparseness. The reason here is the same put forward with word class language models: the probability of occurrence of some word \( w \) in a given context could be very low although the occurrence of the class of \( w \) is quite common in the same context. The same is true of variable length phrases and classes of phrases. The second advantage is a better modelling of long distance dependencies, which are not modelled in a bigram or trigram model. Gathering into classes sequences of words for language modelling is already discussed in [2], [5] and [4]. The distinguishing feature of our approach is the use of syntactic constraints in the definition of word sequences. Other criteria for gathering words into phrases could be used, as in [3].

3 STRUCTURE OF THE LANGUAGE MODEL

The language model is composed of a variable number of local models and a general model, a bigram in our case, which predicts the occurrence probability of a class knowing the preceding class. Each local model gives the probability of some word sequence knowing a class. Formally a local model is represented as a tree-structured weighted finite state automaton (WFSA) which edges are labelled with words. A successful path in the WFSA represents a phrase of the class. Both the bigram and the WFSAs can be put together and regarded as a hidden Markov model\(^1\). Figure 1 shows a subpart of it. In the figure, the probabilities of the external model are denoted by \( P \) while the probabilities of the single local model are denoted by \( P_{AI} \).

The general model being an hidden Markov Model, the probability of a word sequence according to such a model is the sum of the probabilities of all the paths corresponding to the word sequence. There might be several paths since a given word sequence forming a phrase can appear in different classes. Taking into account in the linguistic score of a sentence all its possible segmentations in word sequences of variable length is discussed in [1].

The construction of the model takes place in two stages: the construction of phrase classes and their representation as WFSA, followed by the parameter estimation of the external model. These two stages are described below. The overall process is graphically represented as a block diagram in figure 3.

---

\(*\)This research is supported by France Telecom’s center for research and development (CNET) under the contract 971B427.

\(^1\)The sum of the weights of the transitions exiting a state in each WFSA equals 1.
After the classes have been built, an identifier is associated to every class. Every occurrence of a phrase of a given type (noun phrase, verb phrase, ...) in identical contexts. The context of a phrase is reduced to the $L$ phrases appearing to its left and the $R$ phrases appearing to its right. Every class is therefore associated to a context. A counter is associated to each phrase $P$ of the class $C$, which is incremented every time an occurrence of $P$ appears in the context associated to $C$ in the training corpus. The sum of the counters of the different phrases appearing in a class defines the weight of the class. After the classes have been built those having a weight inferior to a threshold, called the minimum weight threshold, are discarded.

$V$ being the vocabulary of phrases recognized in the corpus, a class can be regarded as a point in the real $|V|$-dimensional space (the $P$-th coordinate of the point being the frequency of the phrase $P$ in the class). An example of verb group class is represented below. It is originally in French, the translation in English was added for readability purpose.

0.69 je voudrais  I want  
0.03 je cherche  I am looking for  
0.03 je voudrais avoir I would like to have  
0.07 je voudrais connaître I would like to know  
0.05 je veux  I want  
0.02 je souhaite avoir I wish to have  
0.04 je recherche  I am looking for  
0.02 je voudrais savoir I would like to know  
0.02 j’aimerais connaître I wish to know

The third step consists in merging together classes having a close internal distribution (frequency of the phrases in the class). The merging is realized by an iterative algorithm which merges together the two closest classes, according to the following distance.

$$d(C_1, C_2) = \sum_{P \in V} |C_1(P) - C_2(P)|$$

Where $C_1(P)$ and $C_2(P)$ denote the frequency of the phrase $P$ in the classes $C_1$ and $C_2$, or the $P$-th coordinate of the points $C_1$ and $C_2$. The algorithm stops when the distance between the two closest classes in superior to a threshold, called the merging threshold. When two classes are merged together, their respective contexts are also merged: if the two classes $C_1$ and $C_2$ are merged to form the class $C_3$, the contexts associated to the class $C_3$ is the union of the contexts of $C_1$ and $C_2$. After the class merging, each class is represented as a weighted finite state automaton which constitutes the local model corresponding to this class. The automaton corresponding to the class above is shown in figure 2. The weights on the transitions have been omitted for the sake of readability.

After the classes have been built, an identifier is associated to every class. Every occurrence of a
phrase $P$ appearing in the partially parsed corpus and belonging to a class $C$ is replaced by the identifier of $C$ when the context of $P$ matches one of the contexts associated to $C$. At the end of this process, a hybrid corpus has been created, composed of words and class identifiers\(^2\). Eventually, a bigram is trained on this corpus using the CMU-Cambridge Statistical Language Modeling Toolkit (www.speech.cs.cmu.edu/speech/SLM_info). This bigram represents the external model.

We have represented in tables 2 and 3 the probabilities given by the model language to a sentence which is quite representative of our corpus. The sentence is *je recherche un emploi à l'étranger* which translates as *I am looking for a job abroad*. The probability given to the sentence by a bigram trained on the complete corpus (referred to as the conventional bigram) is $5.043e^{-07}$ while the probability given by our language model is $3.603e^{-05}$. This increase of the probability is due to the recognition of three phrases in the sentence: the verb phrase *je recherche*, the noun phrase *un emploi* and the prepositional phrase *à l'étranger*. The best path associated to the sentence in the hidden Markov model goes through three automata ($C_{18}$, $C_{33}$, $C_{24}$). The probability of the path is the product of the inter class probabilities (given by the external model) and the intra class probabilities (given by each of the local models). It is higher than the conventional bigram probability because inter class probabilities are globally better than their conventional bigram counterparts, so are the class internal probabilities. We have represented in table 2 the external probabilities ($P(C_{18} | \langle a \rangle)$, $P(C_{33} | C_{18})$ ...). The last column shows the corresponding conventional bigram probabilities, namely, the probability of the first word of the phrase given the last word of the previous phrase ($P(\text{je} | \langle e \rangle)$, $P(\text{un} | \text{recherche})$ ...).

We have represented in the model language to a sentence which is quite representative of our corpus. The sentence is *je recherche un emploi à l'étranger* which translates as *I am looking for a job abroad*. The probability given to the sentence by a bigram trained on the complete corpus (referred to as the conventional bigram) is $5.043e^{-07}$ while the probability given by our language model is $3.603e^{-05}$. This increase of the probability is due to the recognition of three phrases in the sentence: the verb phrase *je recherche*, the noun phrase *un emploi* and the prepositional phrase *à l'étranger*. The best path associated to the sentence in the hidden Markov model goes through three automata ($C_{18}$, $C_{33}$, $C_{24}$). The probability of the path is the product of the inter class probabilities (given by the external model) and the intra class probabilities (given by each of the local models). It is higher than the conventional bigram probability because inter class probabilities are globally better than their conventional bigram counterparts, so are the class internal probabilities. We have represented in table 2 the external probabilities ($P(C_{18} | \langle a \rangle)$, $P(C_{33} | C_{18})$ ...). The last column shows the corresponding conventional bigram probabilities, namely, the probability of the first word of the phrase given the last word of the previous phrase ($P(\text{je} | \langle e \rangle)$, $P(\text{un} | \text{recherche})$ ...).

5 EXPERIMENTS AND RESULTS

The experiments have been conducted CNET’s (France Telecom center for research and develop-

\(^2\)The phrases which have not been replaced by a class identifier are broken down into words.
Table 3 shows the probabilities of each of the three phrases (in bold) given by their corresponding local model and by the conventional bigram, in the last column. The other phrases of each class have been added for reader’s curiosity.

<table>
<thead>
<tr>
<th>class</th>
<th>phrase</th>
<th>local P</th>
<th>bigram P</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_{18}</td>
<td>je voudrais</td>
<td>9.46e-3</td>
<td>5.41e-1</td>
</tr>
<tr>
<td></td>
<td>je cherche</td>
<td>9.05e-1</td>
<td>2.09e-1</td>
</tr>
<tr>
<td></td>
<td>je recherche</td>
<td>8.54e-2</td>
<td>4.00e-2</td>
</tr>
<tr>
<td>C_{33}</td>
<td>un travail</td>
<td>3.14e-2</td>
<td>1.11e-2</td>
</tr>
<tr>
<td></td>
<td>un emploi</td>
<td>9.43e-1</td>
<td>2.89e-1</td>
</tr>
<tr>
<td></td>
<td>un petit-boulot</td>
<td>2.52e-2</td>
<td>6.84e-2</td>
</tr>
<tr>
<td>C_{24}</td>
<td>pour la religion</td>
<td>1.03e-1</td>
<td>1.74e-3</td>
</tr>
<tr>
<td></td>
<td>Midi-Pyrénées</td>
<td>1.03e-1</td>
<td>3.63e-2</td>
</tr>
<tr>
<td></td>
<td>dans la religion parisienne</td>
<td>3.27e-1</td>
<td>2.34e-2</td>
</tr>
<tr>
<td></td>
<td>a2 l’etranger</td>
<td>4.69e-1</td>
<td>7.01e-1</td>
</tr>
</tbody>
</table>

Table 3: Intra class probabilities

Conversely, some sentences which have good bigram probabilities will be given lower probabilities by our language model, as for example the sentence *je recherche un petit-boulot dans la région parisienne* (*I am looking for a job in the region of Paris*). Although the class internal probabilities of the phrases composing this sentence are better than their conventional bigram equivalent, the inter class probabilities showed to be lower explaining the decrease from 3.14e-06 for conventional bigram probability, to 8.04e-07 for our language model.

As described in section 4, several parameters influence the class extraction stage. Some are numerical: the context span, the minimum class weight threshold and the merging threshold, while other are not: the grammar coverage. Different combinations of these parameters give birth to different local models and therefore different language models. Several combinations of the parameters values have been tried that resulted in different perplexity figures, some were lower than the bigram model, other were higher. We did not try yet to optimize the parameters.

The experiments conducted until now showed the importance of two complementary features of the classes. The first is the average length of the phrases in the classes, the second is the class weight after merging. Long phrases will generally be given better probability by local models than by the bigram model, they tend therefore to lower the perplexity. On the other hand long phrases will have low occurrence in the training corpus, they will therefore give rise to light weight classes which will tend to have low probabilities in the external model and therefore increase perplexity. There is hence an important trade-off between average phrase length of a class and its weight. The parameters mentioned above will have different influences on these two variables. Grammar coverage will tend to increase phrase length. Increasing the context span will tend to decrease class weight. Conversely, increasing merging threshold will tend to increase class weight.

6 CONCLUSION

The perplexity decrease, although moderate, shows the potential benefits of combining local language models in the proposed way. Future work will concern optimization of the class construction parameter set, as well as some refinements to the class construction algorithm. These include an iterative algorithm which will iteratively extract classes based on the hybrid corpus (the corpus composed of words and class identifiers) produced by the previous class extraction stage. The result of this algorithm will be to increase the class weight. Other refinements concern smoothing of the local models. Eventually, the integration of this language model in an automatic speech recognition system is planned.

References


