INDIVIDUAL WORD LANGUAGE MODELS AND THE FREQUENCY APPROACH

Elvira I. Sicilia-Garcia, Ji Ming, F. Jack Smith

School Computer Science
Queen’s University of Belfast
Belfast BT7 1NN, Northern Ireland
e.sicilia@qub.ac.uk

ABSTRACT

We present a new method of introducing domain knowledge into an n-gram language model. It is based on a combination of language models for individual word domains. Each word model is built from an individual corpus which is formed by extracting those subsets of the entire training corpus which contain that significant word. When testing, significant words are extracted from a cache and their models are combined with a global language model. Different methods of combining the models are described; one new simple method based on combining frequencies rather than probabilities gives promising results and provides a relatively simple method of introducing domain information into an n-gram language model. A 32% reduction in language model perplexity over the standard 3-gram approach is obtained which is similar to results obtained with other more complex domain models.

1. INTRODUCTION

This paper extends the idea of cache models [1] and trigger models [2] by triggering a new language model for each content word in the cache and combining them with a standard global n-gram language model to produce an improved model.

A training corpus for each significant word is formed from the amalgamation of the text fragments taken from the global training corpus in which that word appears. A significant word is any word which is not a stop word, i.e. not articles, prepositions and some of the most frequently used words in the language such as “will” and common adverbs and adjectives such as “now”, “very”, “some”, etc. The individual language models are then obtained from these corpora. They should be more precise than both the global language model and other domain models. They should also outperform previous cache methods for small caches.

2. THE LANGUAGE MODELS

Our dynamic language model first builds a language model for the global corpus. Frequencies of words and phrases are derived from the corpus and the conditional probability of a word given a sequence of preceding words is estimated. The individual conditional probabilities are approximated by the maximum likelihoods:

\[ P_{ML}(w_j | w_i^{i-1}) = \frac{f(w_j)}{f(w_i^{i-1})} \]

where \( f(X) \) is the frequency of the phrase \( X \) in the text.

In equation (1), there are often unknown sequences of words i.e. phrases which are not in the dictionary; there are also words with low frequency. The maximum likelihood probabilities are then zero or unreliable. In order to improve the prediction of these unseen or unlikely events, and hence the language model, a number of techniques have been explored, for example, the Turing-Good estimate [3], the Katz back-off method [4] or the weighted average (WA) n-gram model [5]. Although any of these or others could be used in our model we use the WA n-gram technique which combines n-gram phrase distributions of several orders using a series of weighting functions. The weighted average probability of a word \( w \) given the preceding words \( w_j \ldots w_m \) is:

\[ P_{WA}(w | w_i^m) = \frac{\mu_0 P_{ML}(w) + \sum_{i=1}^{m} \mu_i P_{ML}(w | w_{m+1-i}^m)}{\sum_{i=0}^{m} \mu_i} \]

where the weighted functions are:

\[ \mu_0 = \ln(T) \text{ and } \mu_i = \ln(f(w_{m+1-i}^m)) \cdot 2^i \]

\( T \) is the number of tokens in the corpus and \( f(w_{m+1-i}^m) \) is the frequency of the sentence \( w_{m+1-i}^m \) in the text.

The unigram maximum likelihood probability of a word is:

\[ P_{ML}(w) = \frac{f(w)}{T} \]
The language model defined by equation (2) and (4) is called here the global language model when trained on the global corpus. Following the creation of the global model comes the creation of a language model for each significant word, which is formed in the same manner as the global language model.

3. PROBABILITY MODELS

We need to combine the probabilities obtained from each word language model and from the global language model, in order to obtain a combined probability for a word given a sequence of words. One simple way to do this is an arithmetic combination of the global language model and the word language models in a linear interpolated model as follows:

\[ P(w|w_1^n) = \lambda_G^iw_i P_{Global}(w|w_1^n) + \sum_{i=1}^{m} \lambda_i P_i(w|w_1^n) \]  

where \( \lambda_G = \sum_{i=1}^{m} \lambda_i = 1 \) and \( P_i(w|w_1^n) \) is the conditional probability in the word language model for the significant word \( w_i \). \( \lambda_i \) is the correspondent weight and \( m \) is the maximum number of word models that we are including.

Ideally the \( \lambda_i \) parameters would be optimised using a held-out training corpus; however this is not practical as we do not know which combination of words \( w_i \) will arise in the cache. So a simpler approach is needed.

3.1. LINEAR INTERPOLATION

A simple way of choosing the \( \lambda \) values is to give the same weight to all the word language models but a different weight to the global language model and put a restriction on the number of word language models to be included. This weighted model is defined as:

\[ P(w|w_1^n) = \alpha P_{Global}(w|w_1^n) + \frac{(1-\alpha)}{m} \sum_{i=1}^{m} P_i(w|w_1^n) \]

and \( \alpha \) and \( m \) are parameters which are chosen to optimise the model.

3.2. EXPONENTIAL DECAY

Furthermore, a method was used based on an exponential decay of the word model probabilities with distance since a word appearing several words previously will generally be less relevant than more recent words. Given a sequence of words, for example, “We had happy times in America…”

\[ P(w|w_1^n) = \left\{ \frac{P_{Global}(w|w_1^n) + P_{Happy}(w|w_1^n) \cdot \exp(-3/d)}{1 + \exp(-3/d) + \exp(-2/d)} \right\} \]

where \( P_{Global}(w|w_1^n) \) is the conditional probability of the word \( w \) following a phrase \( w_1 \cdots w_n \) in the global language model \( P_{Happy}(w|w_1^n) \) is the conditional probability of the word \( w \) following a phrase \( w_1 \cdots w_n \) in the word language model for the significant word \( Happy \). The same definition applies for the word model \( Times \). \( d \) is the exponential decay distance with \( d=5, 10, 15 \), etc. The decaying factor \( \exp(-l/d) \) introduces a cut-off:

\[ \text{if}\ l \geq d \Rightarrow \text{replace} \exp(-ld) \text{by} 0 \]

where \( l \) is the distance from the significant word to the target word and \( d \) is the decay distance.

3.3. WEIGHTED PROBABILITY MODEL

This model is based on the idea that the weight of all of the language models should depend on the size of the training corpora. It is described in the following equation:

\[ P(w|w_1^n) = \frac{\beta_{Global} \cdot P_{Global}(w|w_1^n) + \sum_{i=1}^{m} \beta_i P_i(w|w_1^n)}{\beta_{Global} + \sum_{i=1}^{m} \beta_i} \]

where \( \beta_{Global} \) is the weight for the global language model and \( \beta_i \) is the weight for the word model for the word \( w_i \).

We give more weight to those word models with small training corpus, that is, bigger weights for those significant words that appear less often in the training corpus since they contain more information. The weights used are functions of the size of the word training corpora, that is, the number of tokens of the training corpora \( Ti \). Some examples can be seen in Table 1.

<table>
<thead>
<tr>
<th>We</th>
<th>Had</th>
<th>Happy</th>
<th>Times</th>
<th>In</th>
<th>America</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>


3.4. WEIGHTED EXPONENTIAL MODEL

The weighted exponential language model is a combination of the weighted probability model (3.2) and the exponential decay model (3.3). Given the same example as the one in section 3.3 based on the sentence “We had happy times in America” we define the weighted exponential decay model for the word \( w \),

\[ P(w|w_1^n) = \left\{ \frac{P_{Global}(w|w_1^n) + P_{Happy}(w|w_1^n) \cdot \exp(-1/d)}{1 + \exp(-1/d) + \exp(-1/d)} \right\} \]
where in this case $w$ represents the significant word Spain, as follows:

$$P(w|w^n) = \frac{\beta_{Global} P_{Global}(w|w^n) + \beta_{Happ} P_{Happ}(w|w^n) \exp(-3/d) + \beta_{Time} P_{Time}(w|w^n) \exp(-2/d)}{\beta_{Global} + \beta_{Happ} \exp(-3/d) + \beta_{Time} \exp(-2/d)}$$  \hspace{1cm} (9)$$

where $P(w|w^n)$ is the conditional probability of the word $w$ following a phrase $w_1 \cdots w_n$ in the language model $X$. $d$ is the exponential decay distance with $d=5, 10, 15$, etc. The values of the function $\beta$ are those used before (Table 1).

3.5. UNION MODEL

Another method of combining models is the Probabilistic-Union model [6], which has been effective for noisy speech. This model is based on the logical concept of a disjunction of conjunctions which is implemented as a sum of products. The union model is best illustrated with an example when the number of word models to be included is $m=3$ and if they are assumed to be independent probabilities. The model for $m=3$ has 3 forms:

$$P_{Union}^{(1)} = P_1 \cdot P_2 \cdot P_3$$  \hspace{1cm} (10)$$

$$P_{Union}^{(2)} = P_1 P_2 + P_1 P_3 + P_2 P_3$$  \hspace{1cm} (11)$$

$$P_{Union}^{(3)} = P_1 + P_2 + P_3$$  \hspace{1cm} (12)$$

where $P_{Union}^{(k)} = P_{Union}(w|w^n)$ is the union model of order $k$. $P_i = P_i(w|w^n)$ is the conditional probability for the significant word $w_i$. The combination of the global language model with the probabilistic-union model is defined as follows:

$$P(w|w^n) = \beta_{Global} P_{Global}(w|w^n) + (1-\beta) P_{Union}(w|w^n)$$  \hspace{1cm} (13)$$

The difficulty with the union model is that it requires a great deal of processing to normalize.

4. FREQUENCY MODELS

Instead of combining probabilities to obtain a dynamic language model, it is also possible to combine frequencies before calculating probabilities, i.e. a revised maximum likelihood, replacing equation (1), is:

$$P_{ML}(w_j|w_1^{i-1}) = \frac{\lambda_i f_G(w_1^{i-1}) + \sum_{i=1}^m \lambda_i f_j(w_i^{i-1})}{\lambda_i f_G(w_1^{i-1}) + \sum_{i=1}^m \lambda_i f_j(w_i^{i-1})}$$  \hspace{1cm} (14)$$

This can then be combined using the WA model in equation (2). This simple method is automatically normalised and it is easy to implement and fast to execute. The choice of $\lambda$ is still critical but cannot be optimized beforehand for the same reason that the $\lambda$’s in equation (5) cannot be optimized: we do not know beforehand which combination of words will occur in the cache.

For the frequency model we also combine the frequencies using the same methods than the ones used for probabilities. Note that although the results could be similar to those using the probabilities is important to remember that this new model is easy to compute and automatically normalized.

5. TESTING

The methods described above were compared in some experiments using the Wall Street Journal which contains about 40 million words, and a dictionary of approximately 65,000 words. To compare how the models depend on the size of the training corpus, we test the models in two subsets of the WSJ of 15 million and 5 million word approximately. The test file contains 584 paragraphs, 1869 sentences, 34781 tokens and 3677 words types.

The results reveal a lower perplexity for paragraph context models and 15 million word corpus. This might be due to the limited size of the individual sentence context word corpora, entailing that many of the probabilities will be zero. Therefore the training corpora for the sentence context language models are in general too small to produce enough information to find a good estimate of the individual language model.

6. RESULTS FOR PROBABILITY MODELS

Individual word language models permit an accurate capture of the domains in which significant words occur and hence improve the language model performance. The results indicate that individual word models offer a promising and simple means of introducing domain information into a n-gram language model. The improvement in perplexity so far 28% in Table 2 is similar to that obtained in much more computationally intensive methods based on clustering [8, 9].

Table 2 Improvement in perplexity for different models.

<table>
<thead>
<tr>
<th>Models</th>
<th>3-gram</th>
<th>5-gram</th>
<th>9-gram</th>
<th>Comment.</th>
<th>Best Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Model</td>
<td>0%</td>
<td>19%</td>
<td>20%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear</td>
<td>11%</td>
<td>25%</td>
<td>26%</td>
<td>$\lambda=0.6$</td>
<td>WM=27</td>
</tr>
<tr>
<td>Interpolation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exponential</td>
<td>15%</td>
<td>27%</td>
<td>28%</td>
<td>$\text{Decay}=9$</td>
<td>Cache=60</td>
</tr>
<tr>
<td>Decay Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weighted</td>
<td>14%</td>
<td>27%</td>
<td>28%</td>
<td>$\text{Sqrt(Ti)}$</td>
<td>WM=23</td>
</tr>
<tr>
<td>Probability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weighted</td>
<td>15%</td>
<td>27%</td>
<td>28%</td>
<td>$\text{Decay}=11$</td>
<td>Cache=60</td>
</tr>
<tr>
<td>Exponential</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>-12%</td>
<td>-</td>
<td>-</td>
<td></td>
<td>WM=3</td>
</tr>
<tr>
<td>Union</td>
<td>20%</td>
<td>-</td>
<td>-</td>
<td>$\lambda=0.5$</td>
<td>WM=6</td>
</tr>
</tbody>
</table>
Note that all these improvements are with respect to the 3-gram weighted average language model. Also the union model has not been calculated for n-grams bigger than 3-grams due to their complexity and long processing time.

An improvement of up to 28% has been achieved but using 3 different methods. For the exponential models the optimum is reached when the cache is approximately 60 words, but the same results can be obtained when the cache is reduced to 30 models. For interpolation models, the number of word models to reach the maximum performance is approximately 27 words. Therefore the use of word language models reduces the cache from 500 words [7, 10] to 30 words.

Finally, we test our models using different sizes of n-grams. Although the best results are obtained for 9-gram it is important to observe that the improvement over the 5-gram model is small (1%); using larger database and larger computation time.

7. RESULTS FOR FREQUENCY MODELS

In this article we introduced the new model based in a linear interpolation of frequencies extracted from the global language model with those extracted from the word language models. The language model uses the weighted average model but the maximum likelihood model has been transformed to a maximum likelihood based in frequencies (14). One of the important characteristics when using these frequency models is that the probabilities are normalized. This characteristic makes this model a very quick model to calculate. We present in Table 3 the results obtained for the frequency model.

Table 3 Improvement in perplexity for the different models when applied to the frequencies

<table>
<thead>
<tr>
<th>Model</th>
<th>3-gram</th>
<th>5-gram</th>
<th>Comment</th>
<th>Best Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Interpolation</td>
<td>11%</td>
<td>23%</td>
<td>$\lambda = 0.05$</td>
<td>WM = 16</td>
</tr>
<tr>
<td>Frequency Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exponential Decay Frequency Model</td>
<td>11%</td>
<td>26%</td>
<td>Decay = 70</td>
<td>Cache = 65</td>
</tr>
<tr>
<td>Weighted Frequency Model</td>
<td>25%</td>
<td>32%</td>
<td>$\text{Ln}(T_i)/T_i$</td>
<td>WM = 28</td>
</tr>
<tr>
<td>Weighted exponential</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency Model</td>
<td>25%</td>
<td>32%</td>
<td>Decay = 70</td>
<td>Cache = 70</td>
</tr>
</tbody>
</table>

An improvement of 32% has been achieved using the frequency model. This result is larger than the 28% achieved for the probabilistic model. The weight used for the frequency model is $\text{Ln}(T_i)/T_i$ which is approximately the inverse of the one used for the probabilistic model ($\text{Sqrt}(T_i)$).

We can conclude from these results that the frequency model performs better than the probabilistic model and it is easier to improve and calculate.

8. CONCLUSIONS

In this paper we have introduced the concept of individual word language models to improve language model performance. Individual word language models permit an accurate capture of the domains in which significant words occur and hence improve the language model performance.

The results show that the performance of the word language models depends on the size of the training corpora and so further experiments with bigger corpora are on the way. For the frequency model a 32% improvement has been reached with is important since this new form of combination is simpler and automatically normalized. The results for both methods indicate that the word language models not only are a good method to use in the dynamic language models but they are also easy to manipulate and modify.

9. REFERENCES