Computational Linguistics and Chinese Language Processing
Vol. 15, No. 2, June 2010, pp. 103-126

A Posteriori Individual Word Language Models for Vietnamese Language

Le Quan Ha[*], Tran Thi Thu Van[*], Hoang Tien Long[+],

Nguyen Huu Tinh[+], Nguyen Ngoc Tham[+], and Le Trong Ngoc[*]

Abstract

It is shown that the enormous improvement in the size of disk storage space in recent years can be used to build individual word-domain statistical language models, one for each significant word of a language that contributes to the context of the text. Each of these word-domain language models is a precise domain model for the relevant significant word; when combined appropriately, they provide a highly specific domain language model for the language following a cache, even a short cache. Our individual word probability and frequency models have been constructed and tested in the Vietnamese and English languages. For English, we employed the Wall Street Journal corpus of 40 million English word tokens; for Vietnamese, we used the QUB corpus of 6.5 million tokens. Our testing methods used *a priori* and *a posteriori* approaches. Finally, we explain adjustment of a previously exaggerated prediction of the potential power of *a posteriori* models. Accurate improvements in perplexity for 14 kinds of individual word language models have been obtained in tests, (i) between 33.9% and 53.34% for Vietnamese and (ii) between 30.78% and 44.5% for English, over a baseline global tri-gram weighted average model. For both languages, the best *a posteriori* model is the *a posteriori* weighted frequency model of 44.5% English perplexity improvement and 53.34% Vietnamese perplexity improvement. In addition, five Vietnamese *a posteriori* models were tested to obtain from 9.9% to 16.8% word-error-rate (WER) reduction over a Katz trigram model by the same Vietnamese speech decoder.

[*] Faculty of Information Technology, Hochiminh City University of Industry, Ministry of Industry and Trade, 12 Nguyen Van Bao, Ward 4, Go Vap District, Hochiminh City, Vietnam
E-mail: lequanha@ hui.edu.vn; NLp.Sr@ Shaw.ca; letrongngoc@ hui.edu.vn

[+] Faculty of Information Technology, Nong Lam University of Hochiminh City, Block 6, Linh Trung Ward, Thu Duc District, Hochiminh City, Vietnam
E-mail: {06130155, 06130204, 06130194}@ st.hcmuaf.edu.vn

Keywords: *A Posteriori*, Stop Words, Individual Word Language Models, Frequency Models.

1. Introduction

A human is able to work out the precise domain of a spoken sentence after hearing only a few words. The clear identification of this domain makes it possible for a human to anticipate the following words and combination of words, thus, recognizing speech even in a very noisy environment. This ability to anticipate still cannot be replicated by statistical language models. In this paper, we suggest one way that significant improvement in language modeling performance can be achieved by building domain models for significant words in a language. The word-domain language model extends the idea of cache models (Kuhn & De Mori, 1990) and trigger models (Lau, Rosenfeld, & Roukos, 1993) by triggering a separate *n*-gram language model for each significant word in a cache and combining them to produce a combined model.

The word-domain models are built by collecting a training corpus for each significant word. This is done by amalgamating the text fragments where the word appears in a large global training corpus. In this paper, the text fragments are the sentences containing the significant words. Sicilia-Garcia, Ming, and Smith (2001, 2002) have shown that larger fragments are not needed. We define a significant word as any word that significantly contributes to the context of the text or any word that is (i) not a stop word, *i.e.* not an article, conjunction, or preposition; (ii) not among the most frequently used words in the language, such as “will”; and (iii) not a common adverb or adjective, “now,” “very,” “some,” *etc*.

All other words are considered significant, and a corpus is built for each. A statistical language model is then calculated from this corpus, *i.e.* from all of the sentences containing the word. Therefore, the model should be able to represent the domain of that word. This approach entails a very large number of individual word language models being created, which requires a correspondingly large amount of disk storage; previous experiments by Sicilia-Garcia, Ming, and Smith (2005) were done on twenty thousand individual word language models, which occupied approximately 180 GigaBytes. Thus, this tactic is feasible only given the relatively recent increase in affordable hard disk space. These word models gradually developed from the PhD work of Sicilia-Garcia 1999-2005. Almost at the same time as her research, similar research work was done by Blei, Ng, and Jordan (2003); they originally started from the ideas of Hofmann, Puzicha, and Buhmann (1998) and from Hofmann (1999).

The remaining sections are organized in the following way. First, we discuss the weighted average model our developed approach lies upon. Then, we discuss both the probability models - including linear interpolation and the exponential decay model - and the weighted models, such as the weighted probability model, weighted exponential model, and linear interpolation exponential model with weights. Then, we discuss their corresponding frequency models and we discuss *a priori* and *a posteriori* testing methods. Following this, the corpus, experiments, and results are shown for the probability models, for the frequency models, and for *a posteriori* models. Finally, we provide some conclusions.

2. The Language Models

Experiments had shown that we needed to combine the global language model with the individual word-domain models in order to obtain good results. (This may be due to the limited size of the global corpus in our tests, which was 40 million tokens.) So, we first built a language model for the whole global corpus. Frequencies of words and phrases derived from the corpus and the conditional probability of a word given a sequence of preceding words were calculated. The conditional probabilities were approximated by the maximum likelihoods:

$$P_{ML}\left(w_i \middle| w_1^{i-1}\right) = \frac{f\left(w_1^i\right)}{f\left(w_1^{i-1}\right)} = \frac{f\left(w_1 ... w_{i-1} w_i\right)}{f\left(w_1 ... w_{i-1}\right)} \tag{1}$$

where $f\left(w_1^n\right)$ is the frequency of the phrase $w_1^n = w_1 ... w_{n-1} w_n$ in the text. These probabilities were smoothed by one of the well-known methods, such as Turing-Good estimation (Good, 1953) or the Katz back-off method (Katz, 1987). Although any of these could be used in our experiment to demonstrate the principle of our multiple word-domain models, it was convenient to use the empirical weighted average (WA) linear interpolation *n*-gram model (O'Boyle, Owens & Smith, 1994) because of its simplicity. It gives results comparable to the Katz back-off method but is much quicker to use. The weighted average probability of a word *w* given the preceding words $w_1 \ldots w_{m-1} w_m$ is:

$$P_{WA}\left(w \middle| w_1^m\right) = \frac{\mu_0 P_{ML}\left(w\right) + \sum_{i=1}^{m} \mu_i P_{ML}\left(w \middle| w_{m+1-i}^m\right)}{\sum_{i=0}^{m} \mu_i} \tag{2}$$

where the weighted functions (in the simplest case) are given by

$$\mu_0 = Ln\left(T\right) \quad and \quad \mu_i = Ln\left(f\left(w_{m+1-i}^m\right)\right).2^i \tag{3}$$

where *T* is the number of tokens in the corpus and $f\left(w_{m+1-i}^m\right)$ is the frequency of the sentence $w_{m+1-i} \ldots w_m$ in the text. The unigram maximum likelihood probability of a word is:

$$P_{ML}\left(w\right) = \frac{f\left(w\right)}{T} \tag{4}$$

The language model defined by Equations (2) and (4) is called the global language model when trained on the global corpus. The creation of a language model for each significant word is formed in the same manner as the global language model.

3. Probability Models

We need to combine the probabilities obtained from each word-domain 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 a mathematical combination of the global language model and the word language models in a linear interpolated expression as:

$$P\left(w\middle|w_1^n\right)=\lambda_G P_G\left(w\middle|w_1^n\right)+\sum_{i=1}^{m}\lambda_i P_i\left(w\middle|w_1^n\right) \tag{5}$$

where $\lambda_G+\sum_{1}^{m}\lambda_i=1$ and $P_G\left(w\middle|w_1^n\right)$ is the conditional probability of the word w following a phrase $w_1 \ldots w_{n-1}, w_n$ in the global language model, P_i is the conditional probability in the word language model for the significant word w_i, λ_i is the correspondent weight, and m is the number of word models that are included. Ideally, the λ_i parameters would be optimized 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 method of choosing the λ-values is to give the same weight to all of the word language models but a different weight to the global language model and to put a restriction on the number of word language models to be included. This weighted model is defined as

$$P\left(w\middle|w_1^n\right)=\lambda . P_G\left(w\middle|w_1^n\right)+\frac{(1-\lambda)}{m}\left[\sum_{i=1}^{m}P_i\left(w\middle|w_1^n\right)\right] \tag{6}$$

and λ and m are parameters that are chosen to optimize the model.

3.2 Exponential Decay Model

A method was developed based on an exponential decay of the word model probabilities with distance since a word appearing several words before the target word will generally be less relevant than more recent words. Given a sequence of Vietnamese words, for example, “Tôi đã tri giao với HUI” (meaning “I had friendly relations with HUI”) in Table 1, where 5, 4, 3, 2, 1 represent the distance of the word from the target word “HUI”. The words “tri” (relation) and “giao” (friendly) are significant words for which we have individual word language models.

Table 1. An explanation of distance of words.

Tôi	đã	**tri**	**giao**	với	**HUI**
5	4	3	2	1	

This model for the word *w*, where *w* represents the significant word "HUI," is as follows:

$$P\left(w\middle|w_1^n\right)=\frac{P_G\left(w\middle|w_1^n\right)+P_{tri}\left(w\middle|w_1^n\right).\exp\left(-3/d\right)+P_{giao}\left(w\middle|w_1^n\right).\exp\left(-2/d\right)}{1+\exp\left(-3/d\right)+\exp\left(-2/d\right)} \qquad (7)$$

where $P_G\left(w\middle|w_1^n\right)$ is the conditional probability of the word *w* following a phrase $w_1,w_2\ldots w_n$ in the global language model and $P_{tri}\left(w\middle|w_1^n\right)$ is the conditional probability of the word *w* following a phrase $w_1\ldots w_{n-1}w_n$ in the word language model for the significant word "tri". The same definition applies for the word model "giao". *d* is the exponential decay distance with *d*=5, 10, 15, *etc*. A cache or cut-off is introduced in the model

if $l \geq cache$ => replace exp(-*l*/*d*) by 0

where *l* is the distance from the significant word to the target word.

3.3 Weighted Models

In the two methods above, the weights for the word language models were independent of the size of the word training corpora or the global training corpus. So, we introduced new weights to these models that depend on the size of the training corpora. These weights are functions of the size of the word training corpora, *i.e.* the number of tokens of the training corpora T_i. Examples of the weights can be seen in Table 2.

Table 2. Some of the weights in weighted models.

Weights
$Ln(1+LnT_i)$
$Sqrt(LnT_i)$
LnT_i
$Sqrt(T_i)$
T_i/LnT_i
T_i
T_iLnT_i

An obvious weight to use is a log function to match the information theory, but other weights were also tried. All of these weights are studied in order of weight difference as T_i increases, from the largest difference to the smallest. Their relationship can be seen in Figure 1.

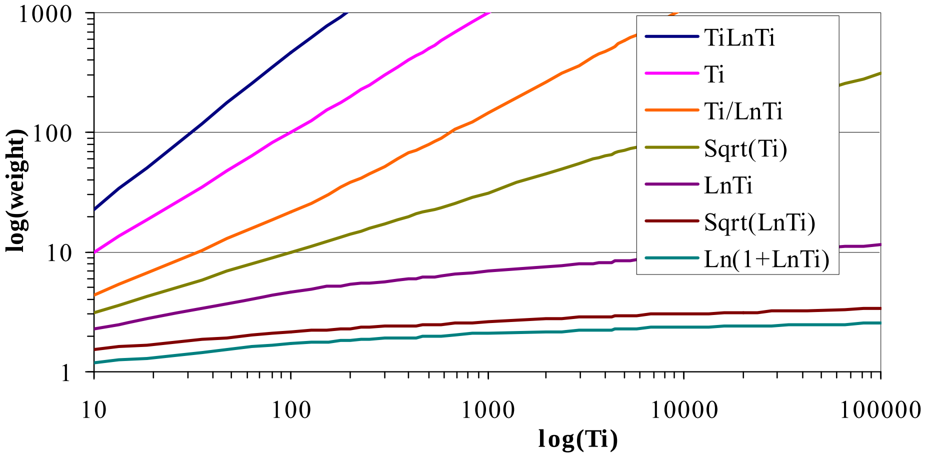


Figure 1. Value of the weights used in these models.

3.3.1 Weighted Probability Model

The weighted probability model is based on the idea that the weight given to a word language model should depend on the size of the training corpora. It is described as follows:

$$P\left(w\middle|w_1^n\right)=\frac{\beta_G.P_G\left(w\middle|w_1^n\right)+\sum_{i=1}^{m}\beta_i.P_i\left(w\middle|w_1^n\right)}{\beta_G+\sum_{i=1}^{m}\beta_i} \tag{8}$$

where β_G is the weight for the global language model and β_i is the weight for the word model for the word w_i. We give more weight to those word models with a small training corpus as they represent models for the less frequent words, which have the most information. The weights used are functions of the size of the word training corpora in Table 2, that is, of the number of tokens of the training corpora T_i.

3.3.2 Weighted Exponential Model

The weighted exponential language model is a combination of the weighted probability model and the exponential decay model. Each language model has two functions: one is the exponential decay, in terms of the distance from the significant word, and the second function is a weight, depending on the size of the word training corpora. We define this model as:

$$P\left(w\middle|w_1^n\right)=\frac{\beta_G.P_G\left(w\middle|w_1^n\right)+\sum_{i=1}^{m}\beta_i.\exp\left(-\frac{x_i}{d}\right)P_i\left(w\middle|w_1^n\right)}{\beta_G+\sum_{i=1}^{m}\beta_i.\exp\left(-\frac{x_i}{d}\right)} \tag{9}$$

where x_i is the distance from the word w_i model.

3.3.3 Linear Interpolation Exponential Model with Weights

Finally, we decided to try another model that is based on a combination of all previous methods. Based on the idea that the global language model could be weighted in a different way from the word language models, the probability of a word given the previous words is:

$$P\left(w\middle|w_1^n\right)=\lambda P_G\left(w\middle|w_1^n\right)+(1-\lambda)\frac{\sum_{i=1}^{m}\beta_i.\exp\left(-\frac{x_i}{d}\right)P_i\left(w\middle|w_1^n\right)}{\sum_{i=1}^{m}\beta_i.\exp\left(-\frac{x_i}{d}\right)} \tag{10}$$

This is equivalent to combining all of the methods seen previously into one model, which we call the linear interpolation exponential model with weights.

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. To do this, replace Equation (1) with:

$$P_{ML}(w_i\mid w_1^{i-1})=\frac{\lambda_G f_G(w_1^i)+\sum_{i=1}^{m}\lambda_i f_i(w_1^i)}{\lambda_G f_G(w_1^{i-1})+\sum_{i=1}^{m}\lambda_i f_i(w_1^{i-1})} \tag{11}$$

This can then be combined using the WA model in Equation (2). This simple method is automatically normalized, and it is easy to implement and fast to execute. The choice of λ is still critical but cannot be optimized from a held out corpus. For the frequency model, we also combine the frequencies using the same methods that are used for probabilities.

4.1 Linear Interpolation Frequency Model

The linear interpolation model applied to the frequency is defined as:

$$f_{FM}\left(w_1^n\right)=f_{FM}\left(w_1w_2...w_n\right)=\lambda.f_G\left(w_1^n\right)+\frac{(1-\lambda)}{m}\left[\sum_{i=1}^{m}f_i\left(w_1^n\right)\right]\quad if\ m>0 \tag{12}$$

where $f_G\left(w_1^n\right)$ is the frequency of the phrase $w_1, w_2\ldots w_n$ from the global model, $f_i\left(w_1^n\right)$ is the frequency of the phrase $w_1, w_2\ldots w_n$ from the word w_i model, and m and λ>0 are chosen parameters.

4.2 Exponential Decay Frequency Model

A method was used based on an exponential decay of the frequencies:

$$f_{FM}\left(w_1^n\right) = f_{FM}\left(w_1 w_2 ... w_n\right) = f_G\left(w_1^n\right) + \sum_{i=1}^{m} \exp\left(-\frac{x_i}{d}\right) f_i\left(w_1^n\right) \tag{13}$$

4.3 Weighted Models

We introduce new weights to these models; they are functions shown in Table 2.

4.3.1 Weighted Frequency Model

The weighted frequency model is a mathematical combination of the word language models with the global language models. The frequency for this model is defined as follows:

$$f_{FM}\left(w_1^n\right) = f_{FM}\left(w_1 w_2 ... w_n\right) = \beta_G f_G\left(w_1^n\right) + \sum_{i=1}^{m} \beta_i . f_i\left(w_1^n\right) \tag{14}$$

4.3.2 Weighted Exponential Decay Frequency Model

As can be seen from the previous section, the frequencies are weighted depending on the size of the training corpora only. In this model, the exponential decaying factor is added to these weights. The new weighted exponential frequency becomes:

$$f_{FM}\left(w_1^n\right) = f_{FM}\left(w_1 w_2 ... w_n\right) = \beta_G f_G\left(w_1^n\right) + \sum_{i=1}^{m} \beta_i . \exp\left(-\frac{x_i}{d}\right) f_i\left(w_1^n\right) \tag{15}$$

5. Testing Methods

Perplexity is a well known measure of the performance of a language model (Jelinek, Mercer, & Bahl, 1983). We calculate the perplexity of each sentence, w_1^n, by

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)...P(w_n \mid w_1^{n-1}) \tag{16}$$

and the perplexity of a sentence by

$$PP(w_1^n) = \exp\left(-\frac{1}{m}\sum_{i=1}^{m} Ln\left(P(w_i \mid w_1 w_2 ... w_{i-1})\right)\right) \tag{17}$$

where m is the number of words in the sentence.

There are two methods of calculating the constituent probabilities on the right hand side of Equation (17) using the word domain language models, one *a priori* by Sicilia-Garcia *et al.* (2001, 2002) and the second *a posteriori,* which was first tried by Sicilia-Garcia *et al.* (2005.)

5.1 *A Priori* Method

In the *a priori* method, we use the global language model and (possibly) individual word models from earlier sentences (*i.e.* from the cache) at the beginning of the sentence because we do not know which significant words are going to appear in the sentence. We then add in a word language model for each significant word after it appears in the sentence. Thus, in the sentence, "The cat sat on the mat," neglecting previous sentences, the first two words are modeled using the global language model, the probability P(sat| the cat) is calculated using the global model combined with the word model for "cat," and the last three words are modeled using the global model combined with the word models for "cat" and "sat".

5.2 *A Posteriori* Method

This, however, is not the only way in which models are tested. For example, in domain language models, researchers extract a whole sentence, paragraph, or document from the test file, find all of the significant words within it, and use all of these words to perform an optimization of the possible domains to find the domain or combination of domains to minimize the perplexity (Seymore, Chen, & Rosenfeld, 1998; Donnelly, 1998; Iyer & Ostendorf, 1999; Sicilia-Garcia *et al.*, 2005.)

To make comparisons with these other domain methods, Sicilia-Garcia *et al.* (2005) also tried this *a posteriori* method to calculate perplexity for word-domain models. To do this, they extracted all of the significant words in a sentence and built a language model based on the global model and the word domain models for the significant words. This was then used to calculate the perplexity of the whole sentence. In the example above, "The cat sat on the mat," the perplexity was calculated using the global model combined with the word domain models for the three words "cat," "sat," and "mat" for the whole sentence calculation. Using this approach, they obtained 68%-69% improvement in perplexity when using the *a posteriori* weighted probability model and the *a posteriori* weighted frequency model. They accepted that this 69% improvement exaggerated the performance but showed the potential power of these *a posteriori* models. Upon further analysis of this, however, we have discovered a slight flaw in the calculation. This can be demonstrated by again considering the example above. They calculated the *a posteriori* probability P(sat| the cat) by employing the global model, the word "cat" model, the word "sat" model, and the word "mat" model. As the unigram "sat," as well as its *n*-grams, obviously occur themselves in every sentence of its own target word "sat" model, this yields an unnaturally high probability P_{sat}(sat| the cat). Hence, the flaw: if we replace the word "sat" by any significant words *xxx* in the test, the probability P_{xxx}(*xxx*| the cat) will also obtain unnaturally high values in its own word *xxx* model.

In this work, we propose and test a corrected method for calculating the *a posteriori* probability P(sat| the cat), which uses only the global model and the word models for "cat" and “mat,” excluding the target word “sat” model. Furthermore, we have tested three different ways of combining word models in our calculation for the linear interpolation probability, the exponential decay probability, and the weighted probability models in the whole sentence test:

i. applying the word models appearing from the beginning until the end of each sentence, excluding the target word model

ii. applying the word models within the phrase history first then the word models that appear later in each sentence, and

iii. ignoring the order of appearance of significant words in each sentence, either existing in the phrase history or occurring later, but applying those significant word models that are located from nearer to farther distances, relative to the target word.

We find out that the (iii) calculation provides 1%-3% better perplexity improvements than (i) and (ii) calculations for all three models. This means that nearer word models supply more reliable probabilities.

Sicilia-Garcia (2002) suggested another whole paragraph calculation with poorer results.

6. Corpus

The methods described above were compared in some Vietnamese and English experiments. For English, we used the Wall Street Journal (WSJ) corpus (Paul & Baker, 1992). Previous research by Sicilia-Garcia *et al.* (2001, 2002, 2005) displayed how the individual word probability models depend on the size of the training corpus for two subsets of the WSJ of approximately 16 million (1988) and 6 million words (1989). The well-known WSJ test file (Paul *et al.*, 1992) contains 584 paragraphs, 1,869 sentences, 34,781 tokens, and 3,677 word types. In this work, we develop these models for the whole combined WSJ corpus of 40 million words. The results reveal a lower perplexity for the larger 40 million word corpus, compared to Sicilia-Garcia *et al.*

For Vietnamese, we employ the syllabic Vietnamese corpus by Ha in 2002 (http://hochiminhcityuniversityofindustry-lequanha.schools.officelive.com/VietnameseQUBCorpus.aspx), with a size of 6,564,445 tokens and 31,402 types. This corpus was collected from popular Vietnamese newspapers, such as The Youth Newspaper, Saigon Liberation Newspaper, Vietnamese Culture and Sports, Motion Pictures Magazine, and Vietnam Express, along with traditional and modern novels and short stories.

Vietnamese is the national and official language of Vietnam. It is the mother tongue of 86% of Vietnam's population, and of about three million overseas Vietnamese people. It is also spoken as a second language by many ethnic minorities of Vietnam. It is part of the Austro-Asiatic language family, of which it has its most speakers, having several times more speakers than the other Austro-Asiatic languages put together. Some Vietnamese vocabulary has been borrowed from Chinese, and it was formerly written using a Chinese-like writing system, albeit in a modified format and with vernacular pronunciation. As a byproduct of French colonial rule, the language displays some influence from French, and the Vietnamese writing system in use today is an adapted version of the Latin alphabet, with additional diacritics for tones and certain letters.

Vietnamese is basically a monosyllabic language with six tones, which gives the language a sing-song effect. A syllable can be repeated with any one of six tones to indicate six different meanings. For example, the syllable "ma" has six different meanings according to the tone this Vietnamese syllable carries - with *level* tone, "ma" means "phantom" or "ghost"; with *low* tone, "mà" means "but," "which," or "who"; with *high rising glottalized* tone, "mã" means "code"; with *dipping-rising* tone, "mả" means "tomb"; with *high rising* tone, "má" means "cheek"; and with *low glottalized* tone, "mạ" means "young rice seedling".

Due to the semantic impact of tonality, we would like to apply our language models for Vietnamese syllables instead of English words.

We also established our Vietnamese test text from the above Vietnamese newspapers, but its content was taken from newspapers of the year 2008, much later than Ha's training text in 2002. Therefore, the Vietnamese test text is totally different from Ha's corpus; it includes 33,108 Vietnamese syllable tokens of 2,226 syllable types within 3,321 sentences.

7. Results

We will first present the results for each *a priori* model, starting from the probability models and moving to the frequency models then show our *a posteriori* results. All perplexity values shown in the tables are accompanied with a percentage, which shows the improvement compared to the global base-line trigram WA model.

7.1 Results for Probability Models

We present the results for all *a priori* probability models, starting from the linear interpolation probability model and going to the linear interpolation exponential decay probability model with weights. It can be seen from the results that the best English and Vietnamese probability model is the weighted exponential probability model, with overall results in Tables 3 and 4.

The best performance achieved using each different type of probability models is shown in Table 5 and Table 6.

The best model is the weighted exponential probability model, with 34% improvement for English and 37% for Vietnamese, while the linear interpolation exponential model with weights - our combination of all of the models - disappointingly only improves 32% for English and 33.9% for Vietnamese. For these models, the number of individual word models required in the cache to reach the maximum performance is 16-23 English words and 29-64 Vietnamese syllables. So, the individual word-domain language model reduces the size of the cache needed from 500 words, as in other models (Clarkson & Robinson, 1997; Donnelly, Smith, Sicilia-Garcia & Ming, 1999), to less than 30 English words or 64 Vietnamese syllables, which is important for spoken language and is closer to the ability of humans.

In Table 5 and Table 6, *WM* represents the number of word language models that is *m* in Equations (6), (8), (9), and (10).

Table 3. The weighted exponential model (English WSJ.)

	Sentence/WSJ		
***n*-gram**	**Perplexity**	**Improvement**	***(d, Cache, Function)***
tri-gram	62.71	16.78%	*(8, 75, Sqrt(1/LnT$_i$))*
5-gram	51.09	32.21%	*(8, 70, LnT$_i$)*
7-gram	49.91	33.77%	*(7, 75, LnT$_i$)*
9-gram	**49.82**	**33.90%**	***(7, 75, LnT$_i$)***

Table 4. The weighted exponential model (Vietnamese QUB.)

	Sentence/QUB		
***n*-gram**	**Perplexity**	**Improvement**	***(d, Cache, Function)***
tri-gram	94.70	22.98%	*(13, 100, Sqrt(LnT$_i$))*
5-gram	80.16	34.81%	*(13, 100, LnT$_i$)*
7-gram	78.12	36.47%	*(13, 100, LnT$_i$)*
9-gram	**77.57**	**36.92%**	***(13, 100, LnT$_i$)***

Table 5. Improvement in perplexity for different probability models, all in sentence contexts (English WSJ.)

Models	**tri-gram**	**9-gram**	**Best Values**
Global	0.00%	26.77%	
Linear Interpolation	11.16%	31.98%	*λ=0.7, WM=23*
Exponential Decay	16.75%	33.72%	*Decay=6, Cache=70*
Weighted Probability	13.90%	32.52%	*WM=16, Sqrt(T$_i$)*
Weighted Exponential	**16.78%**	**33.90%**	***Decay=7, Cache=75, LnT$_i$***
Linear Interpolation Exponential with Weights	12.91%	32.28%	*λ=0.6, Decay=13, Cache=65, Ln(1+LnT$_i$)*

Table 6. Improvement in perplexity for different probability models, all in sentence contexts (Vietnamese QUB.)

Models	tri-gram	9-gram	Best Values
Global	0.00%	20.88%	
Linear Interpolation	18.46%	34.02%	*λ=0.6, WM=64*
Exponential Decay	22.88%	36.54%	*Decay=10, Cache=100*
Weighted Probability	20.45%	34.45%	*WM=29, Sqrt(T_i)*
Weighted Exponential	**22.98%**	**36.92%**	***Decay=13, Cache=100, LnT_i***
Linear Interpolation Exponential with Weights	18.40%	33.88%	*λ=0.6, Decay=57, Cache=100, 1/Ln(1+LnT_i))*

Our weighted exponential model is a special case of the linear interpolation exponential with weights. Hence, it should not have better performance. Sicilia-Garcia *et al.* (2001, 2002, 2005) were also disappointed when this combination model of all other models was not good. The reason for this unusual observation is a conflict occurring between the weighted exponential model (that is a combination of exponential decay and weighted probability models) and the linear interpolation model. The linear interpolation model was optimized on the condition that all significant words are equally treated, while the weighted exponential model treats significant words differently from each other.

In the linear interpolation model, the global "weight" is λ = 0.6 and each Vietnamese syllabic model equally optimized at a "weight" of (1-λ)/m = (1-0.6)/64 = 0.006,25. For the weighted exponential model, however, the global weight is Ln(40M) = 17.5 and a Vietnamese syllabic model with size 10,000 has a weight of Ln(10,000) = 9.21. Another Vietnamese syllabic model with size 100 has its weight of Ln(100) = 4.61. On the linear interpolation exponential with weights, when these weights are multiplied or interpolated together, they break the optimization of both the linear interpolation model and the weighted exponential model; the too small individual "weight" 0.006,25 in the first model and the much larger global weight 17.5 in the latter model are not satisfied.

7.2 Results for Frequency Models

We present the results for each *a priori* frequency model, starting from the linear interpolation frequency model and going to the weighted frequency model. The best English frequency model is the weighted frequency model, and our results for this model are displayed in Table 7. An improvement of 38% has been achieved for the English weighted frequency model.

The best Vietnamese one is the exponential decay frequency model with 47.3% perplexity improvement, as shown in Table 9.

Table 7. The weighted frequency model (English WSJ.)

	Sentence/WSJ		
***n*-gram**	**Perplexity**	**Improvement**	***(WM, Function)***
tri-gram	58.12	22.87%	*(29,1/T_i*Ln(T_i))*
5-gram	47.36	37.16%	*(29, 1/T_i)*
7-gram	**46.70**	**38.03%**	***(29, Ln(T_i)/T_i)***
9-gram	46.73	38.00%	*(29, Ln(T_i)/T_i)*

For the English weighted frequency model, it is important to notice that the perplexity result for the 9-gram model is poorer than the one for the 7-gram language model. We think this is because the word language models in many cases are so small that the 9-gram frequencies are usually zero. In order to recognize or understand a rather short phrase as a tri-gram, its meaning largely depends on its context. The other possibility is that the historical word language models need more weight than the large global model of 40 million tokens. If a significant word language model has 1,000 words in its corpus, then the global weight will approximately be 1/(40M*Ln(40M)) = 1.428E-09 while that word model's weight 1/(1,000*Ln(1,000)) = 0.000,144 is much larger in comparison.

Nevertheless, for longer phrases, such as 5-grams or 9-grams, the meanings of a very long 9-gram is almost obvious by itself and its meaning is less impacted by significant words surrounding it. Sometimes, people understand a long spoken phrase with 9 continuous words without confusion even though they did not hear the previous significant words. Therefore, for longer English phrases, the global weight should gain importance and increase its value, relative to significant words. This means that, in Table 7, the *WM* = 29 for all *n*-grams but, with the tri-gram weight 1/(T_i*Ln(T_i)), is smaller than the 5-gram weight 1/T_i, and 1/T_i is also smaller than Ln(T_i)/T_i of 7-grams and 9-grams.

This is not happening in the Vietnamese weighted frequency model in Table 8 because the value of *WM* is very large, 67. The weight 1/T_i is applied to this Vietnamese model and to the corresponding 5-gram English model in Table 7. For example, we consider the following Vietnamese syllabic 9-gram " Tiếng Việt Nam là một ngôn ngữ đánh vần ," its closest English phrase - "Vietnamese is a syllabic language" - is only a word 5-gram. Therefore, because many English 5-grams will correspond to Vietnamese syllabic 7-grams or 9-grams, this Vietnamese model has the same weights as the 5-gram English weighted frequency model.

Table 8. The weighted frequency model (Vietnamese QUB.)

	Sentence/QUB		
***n*-gram**	**Perplexity**	**Improvement**	***(WM, Function)***
tri-gram	85.88	30.15%	*(67, 1/T_i)*
5-gram	73.55	40.18%	*(67, 1/T_i)*
7-gram	72.63	40.93%	*(67, 1/T_i)*
9-gram	**72.47**	**41.06%**	***(67, 1/T_i)***

Table 9. The exponential decay model (Vietnamese QUB.)

	Sentence/QUB		
***n*-gram**	**Perplexity**	**Improvement**	***(Decay, Cache)***
tri-gram	100.60	18.19%	*(150, 150)*
5-gram	71.68	41.71%	*(150, 145)*
7-gram	66.11	46.24%	*(150, 145)*
9-gram	**64.77**	**47.32%**	***(150, 145)***

The best performance of the frequency models is shown by Table 10 and Table 11.

Table 10. Improvement in perplexity for frequency models, all in sentence contexts (English WSJ.)

Models	**tri-gram**	**9-gram**	**Best Values**
Global	0.00%	26.77%	
Linear Interpolation	15.77%	34.32%	*λ=0.003, WM=29*
Exponential Decay	8.13%	30.78%	*Decay=150, Cache=115*
Weighted Frequency	**22.87%**	**38.00%**	***WM=29, Ln(T_i)/T_i***
Weighted Exponential Decay	22.84%	37.95%	*Decay=100, Cache=85, 1/T_i*

Table 11. Improvement in perplexity for frequency models, all in sentence contexts (Vietnamese QUB.)

Models	**tri-gram**	**9-gram**	**Best Values**
Global	0.00%	20.88%	
Linear Interpolation	22.72%	36.20%	*λ=0.002, WM=46*
Exponential Decay	**18.19%**	**47.32%**	***Decay=150, Cache=145***
Weighted Frequency	30.15%	41.06%	*WM=67, 1/T_i*
Weighted Exponential Decay	30.10%	41.05%	*Decay=100, Cache=100, 1/T_i*

In Table 10 and Table 11, the best model in the Vietnamese corpus is the exponential decay model, which is different from the best in the English corpus because the Vietnamese language is formed by a more limited number of syllables than English words; hence, a Vietnamese syllable has many more different meanings than an English word. Only a significant Vietnamese syllable appearing in the immediate context of a target syllable can change the meaning of the target into a total different topic, but this depends much on the distance from the significant syllable model to the target.

In the weighted frequency models, the best weight in English is (Ln(T_i)/T_i) and Vietnamese (1/T_i). They are different because a Vietnamese sentence is generally longer in syllable length than its corresponding English sentence in word number, for example, the following sentence has 12 syllables “ Tôi làm nghiên cứu về mô hình ngôn ngữ cá thể từ ” and its corresponding English sentence that means “I do research in individual word language models” only contains 8 words.

7.3 Results for *A Posteriori* Models

We investigated our new approach for calculating *a posteriori* probabilities using five models: the linear interpolation probability model, the exponential probability model, the weighted probability model, the linear interpolation frequency model, and the weighted frequency model. We found that the best performance was provided by the *a posteriori* weighted frequency model, which gave a 44.46% English improvement and 53.34% Vietnamese improvement in perplexity. This is better than the performance of much more computationally intensive methods based on clustering (Iyer & Ostendorf, 1999; Clarkson *et al.*, 1997). The *a posteriori* weighted frequency model’s results are displayed in Table 12 and Table 13, and the best results for all different *a posteriori* models are shown in Table 14 and Table 15.

In order to compare to *a priori* models in Table 10, in the *a priori w*eighted frequency model for English, the weights are different, namely Ln(T_i)/T_i in the *a priori* model and 1/T_i*Ln(T_i) in the *a posteriori* model. Besides, the *WM* values, the number of significant word language models or *m* in Equation (14), are quite different, with values of 29 for the *a priori* and 16 for the *a posteriori* models. This can be explained as the concept that people can catch the meaning of a target word more clearly and quickly when they not only hear its previous words but also listen to its following words. Then, the English *a posteriori* model needs to increase weight on significant word models to the global model weight, but it needs to “hear” fewer significant words in the context before it can catch the meaning.

Table 12. The a posteriori weighted frequency model (English WSJ.)

	Whole Sentence Perplexity				**Whole Sentence Improvement**			
Weight	**tri-gram**	**5-gram**	**7-gram**	**9-gram**	**tri-gram**	**5-gram**	**7-gram**	**9-gram**
$T_i*Ln(T_i)$	73.91	56.08	54.61	54.52	1.93%	25.59%	27.53%	27.66%
T_i	73.77	56.00	54.54	54.45	2.11%	25.69%	27.62%	27.75%
$T_i/Ln(T_i)$	73.62	55.92	54.47	54.38	2.31%	25.80%	27.72%	27.85%
$Sqrt(T_i)$	71.88	54.92	53.56	53.48	4.63%	27.13%	28.93%	29.03%
$Ln(T_i)$	67.91	52.52	51.35	51.30	9.88%	30.30%	31.86%	31.93%
$Sqrt(Ln(T_i))$	67.49	52.26	51.10	51.05	10.45%	30.65%	32.19%	32.26%
$Ln(1+Ln(T_i))$	67.34	52.17	51.02	50.97	10.64%	30.77%	32.30%	32.36%
$1/Ln(1+Ln(T_i))$	66.72	51.79	50.66	50.62	11.47%	31.28%	32.78%	32.83%
$Sqrt(1/Ln(T_i))$	66.57	51.69	50.57	50.53	11.67%	31.41%	32.89%	32.95%
$1/Ln(T_i)$	66.07	51.39	50.29	50.25	12.32%	31.81%	33.27%	33.32%
$Sqrt(1/T_i)$	58.31	46.54	45.73	45.72	22.63%	38.24%	39.32%	39.33%
$Ln(T_i)/T_i$	52.00	42.71	42.16	42.19	31.01%	43.32%	44.06%	44.02%
$1/T_i$	51.45	42.44	41.92	41.96	31.73%	43.68%	44.37%	44.32%
$\mathbf{1/T_i*Ln(T_i)}$	**51.09**	**42.30**	**41.81**	**41.85**	**32.21%**	**43.87%**	**44.52%**	**44.46%**

Table 13. The a posteriori weighted frequency model (Vietnamese QUB.)

	Whole Sentence Perplexity				**Whole Sentence Improvement**			
Weight	**tri-gram**	**5-gram**	**7-gram**	**9-gram**	**tri-gram**	**5-gram**	**7-gram**	**9-gram**
$T_i*Ln(T_i)$	122.45	98.83	97.16	96.87	0.41%	19.62%	20.98%	21.22%
T_i	122.27	98.68	97.02	96.73	0.56%	19.74%	21.10%	21.33%
$T_i/Ln(T_i)$	122.03	98.49	96.83	96.54	0.75%	19.90%	21.25%	21.48%
$Sqrt(T_i)$	118.30	95.61	94.06	93.79	3.79%	22.25%	23.51%	23.72%
$Ln(T_i)$	107.45	87.82	86.52	86.30	12.61%	28.58%	29.64%	29.82%
$Sqrt(Ln(T_i))$	106.05	86.82	85.55	85.33	13.75%	29.39%	30.42%	30.60%
$Ln(1+Ln(T_i))$	105.64	86.53	85.27	85.05	14.09%	29.63%	30.65%	30.83%
$1/Ln(1+Ln(T_i))$	103.41	84.94	83.72	83.51	15.90%	30.92%	31.91%	32.08%
$Sqrt(1/Ln(T_i))$	102.96	84.62	83.41	83.20	16.27%	31.19%	32.17%	32.34%
$1/Ln(T_i)$	101.24	83.38	82.21	82.00	17.67%	32.19%	33.14%	33.31%
$Sqrt(1/T_i)$	82.11	69.34	68.46	68.31	33.22%	43.61%	44.32%	44.45%
$Ln(T_i)/T_i$	68.88	59.43	58.75	58.63	43.98%	51.67%	52.22%	52.32%
$1/T_i$	67.31	58.35	57.71	57.59	45.26%	52.55%	53.07%	53.16%
$\mathbf{1/T_i*Ln(T_i)}$	**66.77**	**58.11**	**57.49**	**57.37**	**45.70%**	**52.74%**	**53.25%**	**53.34%**

Table 14. Improvements of a posteriori models (English WSJ.)

***A Posteriori* Models**	**tri-gram**	**9-gram**	**Best Values**
Global	0.00%	26.77%	
Linear Interpolation Probability	30.99%	44.20%	λ*=0.2*
Exponential Decay Probability	28.66%	42.82%	*Decay=100, Cache=5*
Weighted Probability	30.82%	44.17%	$Sqrt(1/Ln(T_i))$
Linear Interpolation Frequency	25.17%	40.66%	λ*=0.001*
Weighted Frequency	**32.21%**	**44.46%**	***WM=16,*** $\boldsymbol{1/T_i*Ln(T_i)}$

Similar to *a posteriori* Vietnamese models in Table 15, the weighted frequency model has *WM*=34 and a weight of $1/(T_i*Ln(T_i))$, while the *a priori* ones are much larger, *WM*=67 and weights $1/T_i$.

Table 15. Improvements of a posteriori models (Vietnamese QUB.)

***A Posteriori* Models**	**tri-gram**	**9-gram**	**Best Values**
Global	0.00%	20.88%	
Linear Interpolation Probability	37.44%	47.63%	λ*=0.3*
Exponential Decay Probability	40.31%	49.28%	*Decay=15, Cache=99*
Weighted Probability	37.62%	47.34%	$Ln(T_i)$
Linear Interpolation Frequency	36.21%	47.03%	λ*=0.001*
Weighted Frequency	**45.70%**	**53.34%**	***WM=34,*** $\boldsymbol{1/T_i*Ln(T_i)}$

Previously, for similar experiments, Sicilia-Garcia *et al.* (2005) reported improvements of 68%-69% for 7-gram models. Nevertheless, that was based on the flawed calculation previously described in the section, *A Posteriori* Method, so the results we present here can be viewed as a true reflection of the performance that can be achieved by these models. Nowadays, with the current condition of PCs, the programming issues of Sicilia-Garcia in training and accessing a language model of 9-grams no longer exist. With a 9-gram phrase length, on the global database of WSJ 4.5 gigabytes, we now complete the *a posteriori* probability linear interpolation model in 30 minutes, while Sicilia-Garcia finished this computer execution for 7-grams in 4 days nonstop. For the *a posteriori* weighted frequency model, we now execute only over 2 days for 9-grams, while Sicilia-Garcia completed 10 full days for 7-grams. (This is without speeding up the algorithms; it is only by upgrading to newer computers.)

The different nature of the Vietnamese syllabic language causes a much smaller proportion of syllables to be significant than English significant words. In a Vietnamese sentence and an English sentence of the same length, there are considerably fewer significant syllables occurring, hence, inconsistent effects are found from our results.

7.4 Speech word-error-rate Results for *A Posteriori* Models

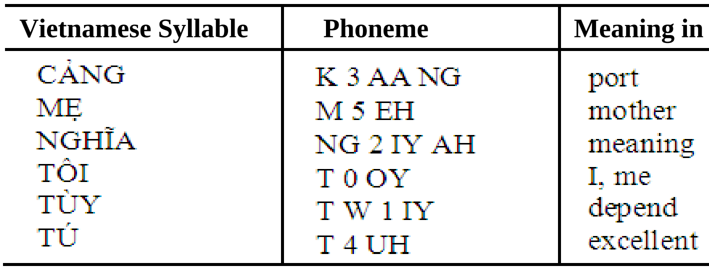
We employ a Vietnamese large vocabulary continuous speech decoder linking to the five Vietnamese *a posteriori* models. Our speech recognition system is syllable-based HMM trained with Vietnamese speech data of 97,850 spoken statements extracted from the Vietnamese QUB text corpus; they are recordings of 326 speech hours by 60 Vietnamese speakers, including 30 men and 30 women from 18 years old to 51 years old. The number of states per HMM is 5, and each state was modeled using a mixture of 16 Gaussians. It is a tone-dependent phoneme model.

7.4.1 Vietnamese VnIPh Lexicon

In March 2009, Ha created the Vietnamese lexicon called Vietnamese International Phonetic Lexicon version 1.0 or VnIPh lexicon (http://hochiminhcityuniversityofindustry-lequanha.schools.officelive.com/VnIPh.aspx).

We employ this lexicon in our speech recognition system; it includes an automatic lexicon generator to create 12,165 Vietnamese syllable entries. We would like to show a few Vietnamese syllables from this lexicon in Table 16.

Table 16. The Vietnamese VnIPh Lexicon (a few entries)



Vietnamese Syllable	Phoneme	Meaning in English
CẢNG	K 3 AA NG	port
MẸ	M 5 EH	mother
NGHĨA	NG 2 IY AH	meaning
TÔI	T 0 OY	I, me
TÙY	T W 1 IY	depend
TÚ	T 4 UH	excellent

In Table 16, phonemes 0, 1, 2, 3, 4, and 5 are the six tones of Vietnamese speech: level, low, high rising glottalized, dipping-rising, high rising, and low glottalized tones. In fact, there are 43 context-independent phonemes: 0, 1, 2, 3, 4, 5, AA, AH, AO, AW, AY, B, CH, D, EH, ER, EY, F, G, HH, IH, IY, K, L, M, N, NG, OW, OY, P, R, S, SH, SIL, T, UH, UW, V, W, WW, WY, Y, and Z.

7.4.2 Vietnamese Speech Tests

We set up our own Vietnamese speech tests that included 21,451 Vietnamese syllables of 3,363 types in 3,834 statements. In Vietnamese speech, the number of syllables is countable because, using the Vietnamese spelling rule, all of the syllables can be obviously formed from the vowels, consonants, and tones. Hence, similar to English words, the number of Vietnamese words, which are combinations of syllables, is uncountable. Nevertheless, all of the possible Vietnamese syllables are fully stored in our lexicon and there are no out-of-vocabulary syllables in our Vietnamese speech tests, even though there are still unknown syllable combinations or unknown Vietnamese words out of the training corpus.

Although the Vietnamese QUB training corpus is clean, without any external noises, our Vietnamese speech tests are noisy with all insertion/deletion/substitution cases of phonemes. Noisy spoken tests are recorded in realistic environments, around cars, fans, and other people talking; each utterance is recorded in a high, medium, or low noise condition.

We apply an unconstrained phone recognizer, and the observed phone sequences are obtained. Our expected phone sequences are constructed using lexicon concatenation based on the Vietnamese syllable transcriptions. In comparison to the expected sequences, our observed sequences show three experimental conditions: well-matched, medium-matched, and highly-mismatched. In our tests, we have three common types of phone errors, which are 16.57% insertion, 18.24% deletion, and 16.21% substitution.

7.4.3 Vietnamese Language Models

Our Vietnamese recognition baseline system uses a Katz back-off trigram language model; hence, we can calculate the word error rate (WER) as follows in Table 17.

All of our Katz probabilities of Vietnamese phrases were stored in a hashed file inside the decoder. In order to link our individual word models into this decoder, we first calculated the combined probabilities of all *n*-grams from unigrams and bigrams up to 9-grams via the five models in Equations (6), (7), (8), (12), and (14). This took us ten days; then, we created five hashed files storing these Vietnamese probabilities with the same structure of the existing Katz hashed file.

In the next step, we replaced the Katz file by each of our five files or our individual word language models. In this way, when speaking, the Vietnamese speech decoder is naturally able to execute on-line. Our Vietnamese word-error-rates are done by tests with direct microphone input from six held-out Vietnamese native speakers, four men and two women, ranging from 21 years old to 52 years old.

Table 17. Noisy Speech A Posteriori WER (Vietnamese QUB.)

***A Posteriori* Models**	**WER**	**Improvement over Baseline**
Baseline	45.65%	
Linear Interpolation Probability	33.67%	11.98%
Exponential Decay Probability	35.78%	9.87%
Weighted Probability	33.33%	12.32%
Linear Interpolation Frequency	33.44%	12.21%
Weighted Frequency	**28.87%**	**16.78%**

8. Conclusions

We have described the concept of using individual word models to improve language model performance. Individual word language models permit an accurate capture of the domains in which significant words occur, thereby improving the model performance. The results indicate that individual word models offer a promising and simple means of introducing domain information into an *n*-gram language model.

Humans probably hear the sounds of several words spoken before using a form of human language model to make a sensible sentence from the sounds, particularly when there are corruptions. Therefore, the idea of using the *a posteriori* method to define the domain might be more appropriate than the *a priori* method. We believe that the use of multiple word-domains, which need only large amounts of relatively inexpensive disk space, models the domain environment of any piece of written or spoken text more accurately than any other domain method. Our Vietnamese word-error-rate measurement to test this theory by linking to a speech decoder is also very good. For our future work, we will apply *a posteriori* language models in automatic speech recognition for English and other languages.

Acknowledgement

We would like to thank Dr. Sicilia-Garcia, Dr. P. Donnelly, Dr. D. Stewart, Dr. P. Hanna, and Prof. F. J. Smith, OBE MRIA. Our heartfelt acknowledgement is sent for the Reviewers of CLCLP. Finally, I would like to appreciate my wife, Mrs. Thuy Vo, and sister, Ms. Giang Vo, for their encouragement to complete this paper during winter in Calgary.

Reference

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research,* 3, 993-022. DOI: 10.1162/jmlr.2003.3.4-5.993.

Clarkson, P. R., & Robinson, A. J. (1997). Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache. *IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2, 799-802. Munich, Germany.

Donnelly, P. (1998). A Domain Based Approach to Natural Language Modelling. *PhD Thesis*. Queen’s University Belfast, Northern Ireland.

Donnelly, P. G., Smith, F. J., Sicilia-Garcia, E. I., & Ming, J. (1999). Language Modelling With Hierarchical Domains. *The 6th European Conference on Speech Communication and Technology,* 4, 1575-1578. Budapest, Hungary.

Good, I. J. (1953). The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika,* 40, 237-254.

Ha, L. Q., Stewart, D. W., Ming, J. & Smith, F. J. (2008). Individual Word Probability Models. *Web Journal of Formal, Computational & Cognitive Linguistics (FCCL),* Issue 10, Russian Association of Artificial Intelligence.

Hofmann, T. (1999). Probabilistic Latent Semantic Indexing. *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99).*

Hofmann, T. (1999). The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data. *IJCAI*, 682-687.

Hofmann, T., Puzicha, J., & Buhmann, J. M. (1998). Unsupervised Texture Segmentation in a Deterministic Annealing Framework. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 803-818.

Iyer, R. M., & Ostendorf, M. (1999). Modeling Long Distance Dependence in Language: Topic Mixture Versus Dynamic Cache Models. *IEEE Transactions on Speech and Audio Processing,* 17(1), 30-39.

Jelinek, F., Mercer, R. L., & Bahl, L. R. (1983). A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 5, 179-190.

Katz, S. M. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser. *IEEE Transactions on Acoustic Speech and Signal Processing,* 35(3), 400-401.

Kuhn, R., & De Mori, R. (1990). A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 12(6), 570-583.

Lau, R., Rosenfeld, R., & Roukos, S. (1993). Trigger-based Language models: A Maximum entropy approach. *IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2, 45-48. Minneapolis, MN.

O'Boyle, P., Owens, M., & Smith, F. J. (1994). Average *n*-gram Model of Natural Language. *Computer Speech and Language,* 8, 337-349.

Paul, D. B., & Baker, J. M. (1992). The Design for the Wall Street Journal-based CSR corpus. *The Second International Conference on Spoken Language Processing*, 899-902.

Seymore, K., Chen, S., & Rosenfeld, R. (1998). Nonlinear Interpolation of Topic Models for Language Model Adaptation. *The 5th International Conference on Spoken Language Processing,* 6, 2503-2506. Sydney, Australia.

Sicilia-Garcia, E. I. (2002). A Study in Dynamic Language Modelling. *PhD Thesis*. Queen's University Belfast, Northern Ireland.

Sicilia-Garcia, E. I., Ming, J., & Smith, F. J. (2001). Triggering Individual Word Domains in *n*-gram Language Models. *The 7th European Conference on Speech Communication and Technology,* 1, 701-704. Aalborg, Denmark.

Sicilia-Garcia, E. I., Ming, J., & Smith, F. J. (2002). Individual Word Language Models and the Frequency Approach. *The 7th International Conference on Spoken Language Processing*, 897-900. Denver, Colorado.

Sicilia-Garcia, E. I., Ming, J., & Smith, F. J. (2005). *A posteriori* multiple word-domain language model. *The 9th European Conference on Speech Communication and Technology (Interspeech-Eurospeech)*, 1285-1288. Lisbon, Portugal.