Word Triggers and the EM Algorithm

Christoph Tillmann and Hermann Ney
Lehrstuhl für Informatik VI
Aachen – University of Technology
D-52074 Aachen, Germany
{tillmann,ney}@informatik.rwth-aachen.de

Abstract

In this paper, we study the use of so-called word trigger pairs to improve an existing language model, which is typically a trigram model in combination with a cache component. A word trigger pair is defined as a long-distance word pair. We present two methods to select the most significant single word trigger pairs. The selected trigger pairs are used in a combined model where the interpolation parameters and trigger interaction parameters are trained by the EM algorithm.

1 Introduction

In this paper, we study the use of so-called word trigger pairs (for short: word triggers) (Bahl et al., 1984, Lau and Rosenfeld, 1993, Tillmann and Ney, 1996) to improve an existing language model, which is typically a trigram model in combination with a cache component (Ney and Essen, 1994).

We use a reference model $p(w|h)$, i.e. the conditional probability of observing the word w for a given history h. For a trigram model, this history h includes the two predecessor words of the word under consideration, but in general it can be the whole sequence of the last M predecessor words. The criterion for measuring the quality of a language model $p(w|h)$ is the so-called log-likelihood criterion (Ney and Essen, 1994), which for a corpus $w_1, ..., w_n, ...w_N$ is defined by:

$$F \quad := \quad \sum_{n=1}^{N} \log p(w_n|h_n),$$

According to this definition, the log-likelihood criterion measures for each position n how well the language model can predict the next word given the knowledge about the preceeding words and computes an average over all word positions n. In the context of language modeling, the log-likelihood criterion F is converted to perplexity PP, defined by $PP := -F/N$.

For applications where the topic-dependence of the language model is important, e.g. text dictation, the history h may reach back several sentences so that the history length M covers several hundred words, say, $M = 400$ as it is for the cache model.

To illustrate what is meant by word triggers, we give a few examples:

airline	......	*flights*
concerto	......	*orchestra*
asks	......	*replies*
neither	......	*nor*
we	......	*ourselves* .

Thus word trigger pairs can be viewed as long-distance word bigrams. In this view, we are faced the problem of finding suitable word trigger pairs. This will be achieved by analysing a large text corpus (i.e. several millions of running words) and learning those trigger pairs that are able to improve the baseline language model. A related approach to capturing long-distance dependencies is based on stochastic variants of link grammars (Pietra and Pietra, 1994).

In several papers (Bahl et al., 1984, Lau and Rosenfeld, 1993, Tillmann and Ney, 1996), selection criteria for *single* word trigger pairs were studied. In this paper, this work is extended as follows:

- **Single-Trigger Model:** We consider the definition of a single word trigger pair. There are two models we consider, namely a backing-off model and a linear interpolation model. For the case of the backing-off model, there is a closed-form solution for estimating the trigger parameter by maximum likelihood. For the linear interpolation model, there is no explicit solution

Christoph Tillmann and Hermann Ney (1997) **Word Triggers and the EM Algorithm.** In T.M. Ellison (ed.) *CoNLL97: Computational Natural Language Learning,* ACL pp 117-124.

anymore, but this model is better suited for the extension towards a large number of simultaneous trigger pairs.

- **Multi-Trigger Model:** In practice, we have to take into account the *interaction* of many trigger pairs. Here, we introduce a model for this purpose. To really use the word triggers for a language model, they must be combined with an existing language model. This is achieved by using linear interpolation between the existing language model and a model for the multi-trigger effects. The parameters of the resulting model, namely the trigger parameters and one interpolation parameter, are trained by the EM algorithm.

- We present experimental results on the Wall Street Journal corpus. Both the single-trigger approach and the multi-trigger approach are used to improve the perplexity of a baseline language model. We give examples of selected trigger pairs with and without using the EM algorithm.

2 Single-Trigger Model

In this section, we review the basic model definition for single word trigger pairs as introduced in (Tillmann and Ney, 1996).

We fix one trigger word pair $(a \rightarrow b)$ and define an extended model $p_{ab}(w|h)$ with an trigger interaction parameter $q(b|a)$. To pave the way for the following extensions, we consider the asymmetric model rather than the symmetric model as originally described in (Tillmann and Ney, 1996).

Backing-Off

As indicated by the results of several groups (Lau and Rosenfeld, 1993, Rosenfeld, 1994, Tillmann and Ney, 1996), the word trigger pairs do not help much to predict the next word if there is already a good model based on specific contexts like trigram, bigram or cache.

Therefore, we allow the trigger interaction $a \rightarrow b$ only if the probability $p(b|h)$ of the reference model is not sufficiently high, i.e. if $p(b|h) < p_0$ for a certain threshold p_0 (note that, by setting $p_0 := 1.0$, the trigger effect is used in *all* cases). Thus, we use the trigger effect only for the following subset of histories:

$$H_{ab} := \{h : a \in h \wedge p(b|h) < p_0\}$$

In the experiments, we used $p_0 := 1.5/W$, where $W = 20\,000$ is the vocabulary size. We define the model $p_{ab}(w|h)$ as an extension of the reference model $p(w|h)$ by a backing-off technique (Katz 87):

$$p_{ab}(w|h) = \begin{cases} q(b|a) & \text{if } h \in H_{ab},\ w = b \\ [1 - q(b|a)] \cdot \dfrac{p(w|h)}{\sum\limits_{w' \neq b} p(w'|h)} & \text{if } h \in H_{ab},\ w \neq b \\ p(w|h) & \text{if } h \notin H_{ab} \end{cases}$$

For a training corpus $w_1 ... w_N$, we consider the log–likelihood functions of both the extended model and the reference model $p(w_n|h_n)$, where we define the history h_n:

$$h_n \quad := \quad w_{n-M}^{n-1} \;=\; w_{n-M} ... w_{n-2} w_{n-1} \quad .$$

For the difference $F_{ab} - F_0$ in the log–likelihoods of the extended language model $p_{ab}(w|h)$ and the reference model $p(w|h)$, we obtain:

$$\begin{aligned} F_{ab} - F_0 &= \\ &= \sum_{n=1}^{N} \log \frac{p_{ab}(w_n|h_n)}{p(w_n|h_n)} \\ &= \sum_{n:\ h_n \in H_{ab}} \log \frac{p_{ab}(w_n|h_n)}{p(w_n|h_n)} \\ &= \sum_{h:\ h \in H_{ab}} \sum_{w} N(h,w)\ \log \frac{p_{ab}(w|h)}{p(w|h)} \\ &= \sum_{h:\ h \in H_{ab}} \Big[N(h,b)\ \log \frac{q(b|a)}{p(b|h)} \\ &\quad + N(h,\bar{b})\ \log \frac{1-q(b|a)}{1-p(b|h)} \Big] \\ &= \tilde{N}(a;b)\ \log q(b|a) \ +\ \tilde{N}(a;\bar{b}) \log[1-q(b|a)] \\ &\quad - \sum_{h:\ h \in H_{ab}} [N(h,b)\ \log p(b|h) \\ &\quad + N(h,\bar{b})\ \log[1-p(b|h)]] \end{aligned}$$

where we have used the usual counts $N(h,w)$:

$$N(h,w) \quad := \sum_{n: h=h_n, w=w_n} 1$$

and two additional counts $\tilde{N}(a;b)$ and $\tilde{N}(a;\bar{b})$ defined particularly for word trigger modeling:

$$\tilde{N}(a;b) \quad := \sum_{h: h \in H_{ab}} N(h,b) \quad = \sum_{n: h_n \in H_{ab}, w_n = b} 1$$

$$\tilde{N}(a;\bar{b}) \quad := \sum_{h: h \in H_{ab}} N(h,\bar{b}) \quad = \sum_{n: h_n \in H_{ab}, w_n \neq b} 1$$

Note that, for the counts $\tilde{N}(a;b)$ and $\tilde{N}(a;\bar{b})$, it does *not* matter how often the triggering word a actually occurred in the history $h \in H_{ab}$.

The unknown trigger parameter $q(b|a)$ is estimated using maximum likelihood estimation. By taking the derivative and setting it to zero, we obtain the estimate:

$$q(b|a) \quad = \quad \frac{\tilde{N}(a;b)}{\tilde{N}(a;b)+\tilde{N}(a;\bar{b})}$$

which can be interpreted as the relative frequency of the occurrence of the word trigger $(a \rightarrow b)$.

Linear Interpolation

Although the backing-off method presented results in a closed-form solution for the trigger parameter $q(b|a)$, the disadvantage is that we have to use an explicit probability threshold p_0 to decide whether or not the trigger effect applies. Furthermore, the ultimate goal is to combine several word trigger pairs into a single model, and it is not clear how this could be done with the backing-off model.

Therefore, we replace the backing-off model by the corresponding model for linear interpolation:

$$p_{ab}(w|h) =$$
$$= \begin{cases} [1-q(b|a)]p(w|h)+\delta(w,b)q(b|a) & \text{if } a \in h \\ & \\ p(w|h) & \text{if } a \notin h \end{cases}$$

$$= \begin{cases} [1-q(b|a)]p(b|h)+q(b|a) & \text{if } a \in h,\ w=b \\ [1-q(b|a)]p(w|h) & \text{if } a \in h,\ w \neq b \\ & \\ p(w|h) & \text{if } a \notin h \end{cases}$$

where $\delta(w,v)=1$ if and only if $v=w$. Note that this interpolation model allows a smooth transition from no trigger effect $(q(b|a) \rightarrow 0)$ to a strong trigger effect $(q(b|a) \rightarrow 1)$.

For a corpus $w_1...w_n...w_N$, we have the log-likelihood difference:

$$\begin{aligned} \Delta F_{ab} \quad = & \sum_{n:a\in h_n, b\neq w_n} \log[1-q(b|a)] \ + \\ & \sum_{n:a\in h_n, b=w_n} \log\left(1-q(b|a)+\frac{q(b|a)}{p(b|h_n)}\right) \\ = \ & [M(a)-N(a;b)]\cdot\log[1-q(b|a)] \ + \\ & \sum_{n:a\in h_n, b=w_n} \log\left(1-q(b|a)+\frac{q(b|a)}{p(b|h_n)}\right) \end{aligned}$$

with the count definition $M(a)$:

$$M(a) \quad := \quad \sum_{n:a\in h_n} 1$$

Thus $M(a)$ counts how many of all positions n $(n=1,\cdots,N)$ with history h_n contain word a and is therefore different from the *unigram* count $N(a)$.

To apply maximum likelihood estimation, we take the derivative with respect to $q(b|a)$ and obtain the following implicit equation for $q(b|a)$ after some elementary manipulations:

$$M(a) \quad = \quad \sum_{n:a\in h_n, b=w_n} \frac{1}{[1-q(b|a)]\cdot p(b|h_n)+q(b|a)}$$

No explicit solution is possible. However, we can give bounds for the exact solution (proof omitted):

$$\frac{\frac{N(a;b)}{M(a)} - <p(b)>}{1-<p(b)>} \quad \leq q(b|a) \leq \quad \frac{N(a;b)}{M(a)},$$

with the definition of the average value $<p(b)>$:

$$<p(b)> \quad = \quad \frac{1}{N(a;b)} \sum_{n:a\in h_n, b=w_n} p(b|h_n)$$

and an additional count $N(a;b)$:

$$N(a;b) \quad := \quad \sum_{n:a\in h_n, b=w_n} 1$$

An improved estimate can be obtained by the EM algorithm (Dempster and Laird, 1977):

$$\bar{q}(b|a) = \\ \frac{1}{M(a)} \sum_{n:a\in h_n, b=w_n} \frac{q(b|a)}{[1-q(b|a)]\cdot p(b|h_n)+q(b|a)}$$

An example of the full derivation of the iteration formula for the EM algorithm will be given in the next section for the more general case of a multi-trigger language model.

3 Multi-Trigger Model

The trigger pairs are used in combination with a conventional baseline model $p(w_n|h_n)$ (e.g. m-gram) to define a trigger model $p_T(w_n|h_n)$:

$$p_T(w_n|h_n) = \\ = \quad (1-\lambda)\cdot p(w_n|h_n)+\frac{\lambda}{M_n}\sum_m \alpha(w_n|w_{n-m})$$

with the trigger parameters $\alpha(w|v)$ that must be normalized for each v:

$$\sum_w \alpha(w|v) = 1 \quad .$$

To simplify the notation, we have used the convention:

$$\sum_m \alpha(w_n|w_{n-m}) \quad = \quad \sum_{m \in \mathcal{M}_n} \alpha(w_n|w_{n-m})$$

with

- $\mathcal{M}_n$: the set of triggering words for position n
- $M_n = |\mathcal{M}_n|$: the number of triggering words for position n

Unfortunately, no method is known that produces closed-form solutions for the maximum-likelihood estimates. Therefore, we resort to the EM algorithm in order to obtain the maximum-likelihood estimates.

The framework of the EM algorithm is based on the so-called $Q(\mu;\overline{\mu})$ function, where $\overline{\mu}$ is the new estimate obtained from the previous estimate μ (Baum, 1972), (Dempster and Laird, 1977). The symbol μ stands for the whole set of parameters to be estimated. The $Q(\mu;\overline{\mu})$ function is an extension of the usual log-likelihood function and is for our model:

$$Q(\cdot) = Q(\{\lambda\}, \{\alpha(w|v)\}; \{\overline{\lambda}\}, \{\overline{\alpha}(w|v)\})$$

$$= \sum_{n=1}^{N} \frac{(1-\lambda)\, p(w_n|h_n) \cdot \log[(1-\overline{\lambda})\, p(w_n|h_n)]}{p_T(w_n|h_n)}$$

$$+ \frac{\frac{\lambda}{M_n} \sum_m \alpha(w_n|w_{n-m}) \cdot \log\left[\frac{\overline{\lambda}}{M_n}\, \overline{\alpha}(w_n|w_{n-m})\right]}{p_T(w_n|h_n)}.$$

Taking the partial derivatives and solving for $\overline{\lambda}$, we obtain:

$$\overline{\lambda} = \frac{1}{N} \sum_{n=1}^{N} \frac{\frac{\lambda}{M_n} \sum_m \alpha(w_n|w_{n-m})}{(1-\lambda)p(w_n|h_n) + \frac{\lambda}{M_n} \sum_m \alpha(w_n|w_{n-m})}$$

When taking the partial derivatives with respect to $\overline{\alpha}(w|v)$, we use the method of Lagrangian multipliers for the normalization constraints and obtain:

$$\overline{\alpha}(w|v) \quad = \quad \frac{A(w,v)}{\sum_{w'} A(w',v)} \quad \text{with}$$

$$A(w,v) = \alpha(w|v) \cdot$$

$$\sum_{n=1}^{N} \delta(w,w_n) \frac{\frac{\lambda}{M_n} \sum_m \delta(v, w_{n-m})}{(1-\lambda)p(w_n|h_n) + \frac{\lambda}{M_n} \sum_m \alpha(w_n|w_{n-m})}$$

Note how the interaction of word triggers is taken into account by a local weighting effect: For a fixed position n with $w_n = w$, the contribution of a particular observed *distant* word pair $(v...w)$ to $\overline{\alpha}(w|v)$ depends on the interaction parameters of *all other* word pairs $(v'...w)$ with $v' \in \{w_{n-M}^{n-1}\}$ *and* the baseline probability $p(w|h)$.

Note that the local convergence property still holds when the length M_n of the history is dependent on the word position n, e.g. if the history reaches back only to the beginning of the current paragraph.

A remark about the functional form of the multi-trigger model is in order. The form chosen in this paper is a sort of linear combination of the trigger pairs. A different approach is to combine the various trigger pairs in multiplicative way, which results from a Maximum-Entropy approach (Lau and Rosenfeld, 1993).

4 Experimental results

Language Model Training and Corpus

We first describe the details of the language model used and of its training. The trigger pairs were selected as described in subsection 2 and were used to extend a baseline language model. As in many other systems, the baseline language model used here consists of two parts, an m–gram model (here: trigram/bigram/unigram) and a cache part(Ney and Essen, 1994). Since the cache effect is equivalent to self-trigger pairs $(a \to a)$, we can expect that there is some trade-off between the word triggers and the cache, which was confirmed in some initial informal experiments.

For this reason, it is suitable to consider the *simultaneous* interpolation of these three language model parts to define the refined language model. Thus we have the following equation for the refined language model $p(w_n|h_n)$:

$$p(w_n|h_n) =$$
$$\lambda_M \cdot p_M(w_n|h_n) + \lambda_C \cdot p_C(w_n|h_n) + \lambda_T \cdot p_T(w_n|h_n),$$

where $p_M(w_n|h_n)$ is the m–gram model, $p_C(w_n|h_n)$ is the cache model and $p_T(w_n|h_n)$ is the trigger model. The three interpolation parameters must be normalized:

$$\lambda_M + \lambda_C + \lambda_T \quad = \quad 1 \quad .$$

The details of the m–gram model are similar to those given in (Ney and Generet, 1995). The cache model $p_C(w_n|w_{n-M}^{n-1})$ is defined as:

$$p_C(w_n|w_{n-M}^{n-1}) \quad = \quad \frac{1}{M} \sum_{m=1}^{M} \delta(w_n, w_{n-m}) \quad ,$$

Table 1: Effect of word trigger on test set perplexity (a) and interpolation parameter $\lambda_M, \lambda_C, \lambda_T$ (b).

		training corpus		
a)	language model	1 Mio	5 Mio	39 Mio
	trigram with no cache	255.1	168.4	104.9
	trigram with cache	200.0	138.9	92.1
	+ triggers: no EM	183.2	129.8	88.5
	+ triggers: with EM	179.0	127.2	87.4
b)	+ triggers: no EM	.83 / .11 / .06	.86 / .09 / .05	.89 / .08 / .04
	+ triggers: with EM	.82 / .10 / .09	.85 / .09 / .07	.86 / .07 / .07

where $\delta(w, v) = 1$ if and only if $v = w$. The trigger model $p_T(w_n|h_n)$ is defined as:

$$p_T(w_n|w_{n-M}^{n-1}) \quad = \quad \frac{1}{M}\sum_{m=1}^{M} \alpha(w_n|w_{n-m}) \quad .$$

There were two methods used to compute the trigger parameters:

- **method 'no EM'**: The trigger parameters $\alpha(w|v)$ are obtained by renormalization from the single trigger parameters $q(w|v)$:

$$\alpha(w|v) \quad = \quad \frac{q(w|v)}{\sum_{w'} q(w'|v)}$$

  The backing-off method described in Section 2.1 was used to select the top-K most significant single trigger pairs. In the experiments, we used $K = 1.5$ million trigger pairs.

- **method 'with EM'**: The trigger parameters $\alpha(w|v)$ are initialized by the 'no EM' values and re-estimated using the EM algorithm as described in Section 3. The typical number of iterations is 10.

The experimental tests were performed on the Wall Street Journal (WSJ) task (Paul and Baker, 1992) for a vocabulary size of 20000 words. To train the m–gram language model and the interpolation parameters, we used three training corpora with sizes of 1, 5 and 39 million running words. However, the word trigger pairs were *always* selected and trained from the 39-million word training corpus. In the experiments, the history h was defined to start with the most recent article delimiter.

The interpolation parameters are trained by using the EM algorithm. In the case of the 'EM triggers', this is done jointly with the reestimation of the trigger parameters $\alpha(w|v)$. To avoid the overfitting of the interpolation parameters on the training corpus, which was used to train *both* the m–gram language model *and* the interpolation parameters, we applied the leaving-one-out technique.

Examples of Trigger Pairs

In Table 2 and Table 3 we present examples of selected trigger pairs for the two methods no EM and EM. For a fixed triggering word v, we show the most significant triggered words w along with the trigger interaction parameter $\alpha(w|v)$ for both methods. There are 8 triggering words v for each of which we show the 15 triggered words w with the highest trigger parameter $\alpha(w|v)$. The triggered words w are sorted by the $\alpha(w|v)$ parameter. ¿From the table it can be seen that for the no EM trigger pairs the trigger parameter $\alpha(w|v)$ varies only slightly over the triggered words w. This is different for the EM triggers, where the trigger parameters $\alpha(w|v)$ have a much larger variation. In addition the probability mass of the EM-trained trigger pairs is much more concentrated on the first 15 triggered words.

Perplexity Results

The perplexity was computed on a test corpus of 325 000 words from the WSJ task. The results are shown in Table 1 for each of the three training corpora (1,5 and 39 million words). For comparison purposes, the perplexities of the trigram model with and without cache are included. As can be seen from this table, the trigger model is able to improve the perplexities in all conditions, and the EM triggers are consistently (although sometimes only slightly) better than the no EM triggers. There is an effect of the training corpus size: if the trigram model is already well trained, the trigger model does not help as much as for a less well trained trigram model. This observation is confirmed by the part b of Table 1, which shows the EM trained interpolation parameters. As the size of the training corpus decreases the relative weight of the cache and trigger component increases. Furthermore in the last row of Table 1 it can be seen that the relative weight of the trigger component increases after the EM training which indicates that the parameters of our trigger modell are successfully trained by this EM approach.

Table 2: Triggered words w along with $\alpha(w \mid v)$ for triggering word v.

$v =$ "**added**"				$v =$ "**airlines**"			
no EM		with EM		no EM		with EM	
w	$\alpha(w \mid v)$	w	$\alpha(w \mid v)$	w	$\alpha(w \mid v)$	w	$\alpha(w \mid v)$
declining	0.011	declined	0.106	passengers	0.015	airline	0.296
adding	0.010	asked	0.080	carriers	0.013	air	0.064
Bayerische	0.010	estimated	0.070	passenger	0.013	Continental	0.056
positive	0.009	asserted	0.055	United's	0.013	carrier	0.049
speculation	0.009	dropped	0.049	Trans	0.012	carriers	0.046
concerns	0.009	concerns	0.036	Continental's	0.011	passengers	0.037
finished	0.008	conceded	0.033	Eastern's	0.010	flight	0.035
remaining	0.008	adding	0.029	flights	0.010	United	0.032
reporting	0.008	recommended	0.028	fare	0.009	flights	0.029
confusion	0.008	contended	0.023	airline	0.009	Delta	0.026
excess	0.007	confusion	0.023	American's	0.009	fares	0.024
falling	0.007	reporting	0.020	pilots'	0.008	Eastern	0.023
disappointing	0.007	adequate	0.017	airlines'	0.008	carrier's	0.020
eased	0.007	referring	0.016	travel	0.008	frequent	0.018
equities	0.007	contributed	0.016	planes	0.008	passenger	0.018

$v =$ "**business**"				$v =$ "**buy**"			
no EM		with EM		no EM		with EM	
w	$\alpha(w \mid v)$	w	$\alpha(w \mid v)$	w	$\alpha(w \mid v)$	w	$\alpha(w \mid v)$
competitors	0.004	corporate	0.146	purchases	0.005	purchase	0.145
changing	0.004	businesses	0.102	acquiring	0.005	buying	0.092
creative	0.004	marketing	0.056	privately	0.005	purchases	0.051
simply	0.004	customers	0.047	deals	0.004	well	0.050
deals	0.004	computer	0.026	speculative	0.004	bought	0.042
competing	0.004	executives	0.024	partly	0.004	cash	0.030
hiring	0.004	working	0.023	financing	0.004	deal	0.028
Armonk	0.004	competitive	0.022	huge	0.004	potential	0.026
personnel	0.004	manufacturing	0.019	immediately	0.004	future	0.025
businesses	0.003	product	0.018	aggressive	0.004	couldn't	0.024
faster	0.003	profits	0.017	declining	0.004	giving	0.022
offices	0.003	corporations	0.016	borrowing	0.004	buys	0.019
inventory	0.003	started	0.015	cheap	0.004	together	0.018
successful	0.003	businessmen	0.014	cyclical	0.004	bid	0.018
color	0.003	offices	0.011	investing	0.004	buyers	0.017

Table 3: Triggered words w along with $\alpha(w|v)$ for triggering word v.

v = "**company**"				v = "**Ford**"			
no EM		with EM		no EM		with EM	
w	$\alpha(w\|v)$	w	$\alpha(w\|v)$	w	$\alpha(w\|v)$	w	$\alpha(w\|v)$
adding	0.002	management	0.092	Ford's	0.039	Ford's	0.651
acquiring	0.002	including	0.037	Dearborn	0.020	auto	0.063
publicly	0.001	top	0.028	Chrysler's	0.014	Dearborn	0.056
depressed	0.001	employees	0.027	Chevrolet	0.013	Chrysler	0.028
financially	0.001	will	0.024	Lincoln	0.013	Mercury	0.022
roughly	0.001	plans	0.018	truck	0.012	Taurus	0.021
prior	0.001	unit	0.017	Mazda	0.011	Mustang	0.013
reduced	0.001	couldn't	0.017	vehicle	0.010	Escort	0.011
overseas	0.001	hasn't	0.016	Dodge	0.009	Lincoln	0.010
remaining	0.001	subsidiary	0.014	incentive	0.009	Tempo	0.009
competitors	0.001	previously	0.014	Buick	0.009	parts	0.007
substantially	0.001	now	0.013	dealer	0.008	car	0.006
rival	0.001	since	0.011	vans	0.008	pattern	0.006
partly	0.001	won't	0.011	car's	0.008	Henry	0.006
privately	0.001	executives	0.011	Honda	0.008	Jaguar	0.006
v = "**love**"				v = "**says**"			
w	$\alpha(w\|v)$	w	$\alpha(w\|v)$	w	$\alpha(w\|v)$	w	$\alpha(w\|v)$
characters	0.006	human	0.051	deep	0.002	adds	0.090
physical	0.005	lovers	0.044	changing	0.002	low	0.053
turns	0.005	passion	0.039	starting	0.002	suggests	0.031
beautiful	0.005	turns	0.031	simply	0.002	concedes	0.024
comic	0.005	beautiful	0.030	tough	0.002	explains	0.019
playing	0.005	spirit	0.029	dozens	0.002	contends	0.017
fun	0.005	marriage	0.020	driving	0.002	notes	0.016
herself	0.005	phil	0.019	twice	0.002	agrees	0.016
rock	0.005	lounge	0.017	experts	0.002	thinks	0.015
stuff	0.005	dresses	0.017	cheap	0.002	insists	0.015
dance	0.004	stereotype	0.016	winning	0.002	get	0.014
evil	0.004	wonder	0.015	minor	0.002	hot	0.013
God	0.004	songs	0.015	critics	0.002	early	0.013
pain	0.004	beautifully	0.014	nearby	0.002	sees	0.012
passion	0.004	muscular	0.014	living	0.002	consultant	0.012

5 Conclusions

We have presented a model and an algorithm for training a multi-word trigger model along with some experimental evaluations. The results can be summerized as follows:

- The trigger parameters for all word triggers are jointly trained using the EM algorithm. This leads to a systematic (although small) improvement over the condition that each trigger parameter is trained separately.
- The word-trigger model is used in combination with a full language model (m-gram /cache) . Thus the perplexity is reduced from 138.9 to 127.2 for the 5-million training corpus and from 92.2 to 87.4 for the 39-million corpus.

References

L. E. Baum. 1972. "An Inequality and Associated Maximization Technique in Statistical Estimation of a Markov Process", Inequalities, Vol. 3, No. 1, pp. 1-8.

L. R. Bahl, F. Jelinek, R. L. Mercer, A. Nadas. 1984. "Next Word Statistical Predictor", IBM Tech. Disclosure Bulletin, Vol. 27, No. 7A, pp. 3941-42, December.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer. 1993. "Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics*, Vol. 19, No. 2, pp. 263-311, June.

A. P. Dempster, N. M. Laird, D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm", J. Royal Statist. Soc. Ser. B (methodological), Vol. 39, pp. 1-38.

S.M. Katz. 1993. "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", in IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 35, pp. 400–401, March.

R. Lau, R. Rosenfeld, S. Roukos. 1993. "Trigger-Based Language Models: A Maximum Entropy Approach", in Proc. IEEE Inter. Conf. on Acoustics, Speech and Signal Processing, Minneapolis, MN, Vol. II, pp. 45-48, April.

H. Ney, U. Essen, R. Kneser. 1994. "On Structuring Probabilistic Dependencies in Language Modeling", *Computer Speech and Language*, Vol. 8, pp. 1-38.

H. Ney, M. Generet, F. Wessel. 1995. "Extensions of Absolute Discounting for Language Modeling", in Proc. Fourth European Conference on Speech Communication and Technology, Madrid, pp. 1245-1248, September.

D.B. Paul and J.B. Baker. 1992. "The Design for the Wall Street Journal-based CSR Corpus", in Proc. of the DARPA SLS Workshop, pp. 357–361, February.

S. Della Pietra, V. Della Pietra, J. Gillett, J. Lafferty, H. Printz and L. Ures. 1994. "Inference and Estimation of a Long-Range Trigram Model", in *Lecture Notes in Artificial Intelligence*, Grammatical Inference and Applications, ICGI-94, Alicante, Spain, Springer-Verlag, pp. 78-92, September.

R. Rosenfeld. 1994. "Adaptive Statistical Language Modeling: A Maximum Entropy Approach", *Ph.D. thesis*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, CMU–CS–94–138.

C. Tillmann and H. Ney. 1996. "Selection Criteria for Word Triggers in Language Modeling". , in *Lecture Notes in Artificial Intelligence*, Int. Colloquium on Grammatical Inference, Montpellier, France, Springer-Verlag, pp. 95-106, September.