Tied–Mixture Language Modeling in Continuous Space

Ruhi Sarikaya
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
sarikaya@us.ibm.com

Mohamed Afify
Orange Labs.
Cairo, Egypt
mohamed_afify2001@yahoo.com

Brian Kingsbury
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
bedk@us.ibm.com

Abstract

This paper presents a new perspective to the language modeling problem by moving the word representations and modeling into the continuous space. In a previous work we introduced Gaussian-Mixture Language Model (GMLM) and presented some initial experiments. Here, we propose Tied-Mixture Language Model (TMLM), which does not have the model parameter estimation problems that GMLM has. TMLM provides a great deal of parameter tying across words, hence achieves robust parameter estimation. As such, TMLM can estimate the probability of any word that has as few as two occurrences in the training data. The speech recognition experiments with the TMLM show improvement over the word trigram model.

1 Introduction

Despite numerous studies demonstrating the serious short-comings of the n–gram language models, it has been surprisingly difficult to outperform n–gram language models consistently across different domains, tasks and languages. It is well-known that n–gram language models are not effective in modeling long range lexical, syntactic and semantic dependencies. Nevertheless, n–gram models have been very appealing due to their simplicity; they require only a plain corpus of data to train the model. The improvements obtained by some more elaborate language models (Chelba & Jelinek, 2000; Erdogan et al., 2005) come from the explicit use of syntactic and semantic knowledge put into the annotated corpus.

In addition to the mentioned problems above, traditional n–gram language models do not lend themselves easily to rapid and effective adaptation and discriminative training. A typical n–gram model contains millions of parameters and has no structure capturing dependencies and relationships between the words beyond a limited local context. These parameters are estimated from the empirical distributions, and suffer from data sparseness. n–gram language model adaptation (to new domain, speaker, genre and language) is difficult, simply because of the large number of parameters, for which large amount of adaptation data is required. Instead of updating model parameters with an adaptation method, the typical practice is to collect some data in the target domain and build a domain specific language model. The domain specific language model is interpolated with a generic language model trained on a larger domain independent data to achieve robustness. On the other hand, rapid adaptation for acoustic modeling, using such methods as Maximum Likelihood Linear Regression (MLLR) (Legetter & Woodland, 1995), is possible using very small amount of acoustic data, thanks to the inherent structure of acoustic models that allow large degrees of parameter tying across different words (several thousand context dependent states are shared by all the words in the dictionary). Likewise, even though discriminatively trained acoustic models have been widely used, discriminatively trained languages models (Roark et al., 2007) have not widely accepted as a standard practice yet.

In this study, we present a new perspective to the language modeling. In this perspective, words are not treated as discrete entities but rather vectors of real numbers. As a result, long–term semantic relationships between the words could be quantified and can be integrated into a model. The proposed formulation casts the language modeling problem as

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL, pages 459–467,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

an acoustic modeling problem in speech recognition. This approach opens up new possibilities from rapid and effective adaptation of language models to using discriminative acoustic modeling tools and methods, such as Minimum Phone Error (MPE) (Povey & Woodland, 2002) training to train discriminative language models.

We introduced the idea of language modeling in continuous space from the acoustic modeling perspective and proposed Gaussian Mixture Language Model (GMLM) (Afify et al., 2007). However, GMLM has model parameter estimation problems. In GMLM each word is represented by a specific set of Gaussian mixtures. Robust parameter estimation of the Gaussian mixtures requires hundreds or even thousands of examples. As a result, we were able to estimate the GMLM probabilities only for words that have at least 50 or more examples. Essentially, this was meant to estimate the GMLM probabilities for only about top 10% of the words in the vocabulary. Not surprisingly, we have not observed improvements in speech recognition accuracy (Afify et al., 2007). Tied-Mixture Language Model (TMLM) does not have these requirements in model estimation. In fact, language model probabilities can be estimated for words having as few as two occurrences in the training data.

The concept of language modeling in continuous space was previously proposed (Bengio et al., 2003; Schwenk & Gauvain, 2003) using Neural Networks. However, our method offers several potential advantages over (Schwenk & Gauvain, 2003) including adaptation, and modeling of semantic dependencies because of the way we represent the words in the continuous space. Moreover, our method also allows efficient model training using large amounts of training data, thanks to the acoustic modeling tools and methods which are optimized to handle large amounts of data efficiently.

It is important to note that we have to realize the full potential of the proposed model, before investigating the potential benefits such as adaptation and discriminative training. To this end, we propose TMLM, which does not have the problems GMLM has and, unlike GMLM we report improvements in speech recognition over the corresponding n–gram models.

The rest of the paper is organized as follows. Section 2 presents the concept of language modeling in continuous space. Section 3 describes the tied–mixture modeling. Speech recognition architecture is summarized in Section 4, followed by the experimental results in Section 5. Section 6 discusses various issues with the proposed method and finally, Section 7 summarizes our findings.

2 Language Modeling In Continuous Space

The language model training in continuous space has three main steps; namely, creation of a co–occurrence matrix, mapping discrete words into a continuous parameter space in the form of vectors of real numbers and training a statistical parametric model. Now, we will describe each step in detail.

2.1 Creation of a co–occurrence Matrix

There are many ways that discrete words can be mapped into a continuous space. The approach we take is based on Latent Semantic Analysis (LSA) (Deerwester et al., 1990), and begins with the creation of a co–occurrence matrix. The co–occurrence matrix can be constructed in several ways, depending on the morphological complexity of the language. For a morphologically impoverished language, such as English the co–occurrence matrix can be constructed using word bigram co–occurrences. For morphologically rich languages, there are several options to construct a co–occurrence matrix. For example, the co–occurrence matrix can be constructed using either words (word–word co–occurrences) or morphemes (morpheme–morpheme co–occurrences), which are obtained after morphologically tokenizing the entire corpus. In addition to word–word or morpheme–morpheme co–occurrence matrices, a word–morpheme co–occurrence matrix can also be constructed. A word w can be decomposed into a set of prefixes, stem and suffixes: $w = [pfx_1 + pfx_2 + pfx_n + stem + sfx_1 + sfx_2 + sfx_n]$. The columns of such a matrix contain words and the rows contain the corresponding morphological decomposition (i.e. morphemes) making up the word. The decomposition of this matrix (as will be described in the next sub-section) can allow joint modeling of words and morphemes in one model.

In this study, we use morpheme level bigram co–occurrences to construct the matrix. All the morpheme[1] bigrams are accumulated for the entire corpus to fill in the entries of a co–occurrence matrix, C, where $C(w_i, w_j)$ denotes the counts for which word w_i follows word w_j in the corpus. This is a large, but very sparse matrix, since typically a small number of words follow a given word. Because of its large size and sparsity, Singular Value Decomposition (SVD) is a natural choice for producing a reduced-rank approximation of this matrix.

The co–occurrence matrices typically contain a small number of high frequency events and a large number of less frequent events. Since SVD derives a compact approximation of the co–occurrence matrix that is optimal in the least–square sense, it best models these high-frequency events, which may not be the most informative. Therefore, the entries of a word-pair co–occurrence matrix are smoothed according to the following expression:

$$\hat{C}(w_i, w_j) = \log(C(w_i, w_j) + 1) \tag{1}$$

Following the notation presented in (Bellegarda, 2000) we proceed to perform the SVD as follows:

$$\hat{C} \approx USV^T \tag{2}$$

where U is a left singular matrix with row vectors u_i $(1 \leq i \leq M)$ and dimension $M \times R$. S is a diagonal matrix of singular values with dimension $R \times R$. V is a right singular matrix with row vectors v_j $(1 \leq j \leq N)$ and dimension $N \times R$. R is the order of the decomposition and $R \ll min(M, N)$. M and N are the vocabulary sizes on the rows and columns of the co–occurrence matrix, respectively. For word–word or morpheme–morpheme co–occurrence matrices $M = N$, but for word–morpheme co–occurrence matrix, M is the number of unique words in the training corpus and N is the number of unique morphemes in morphologically tokenized training corpus.

2.2 Mapping Words into Continuous Space

The continuous space for the words listed on the rows of the co–occurrence matrix is defined as the space spanned by the column vectors of $A_{M \times R} = US$. Similarly, the continuous space for the words on the columns are defined as the space spanned by the row vectors of $B_{R \times N} = SV^T$. Here, for a word–word co–occurrence matrix, each of the scaled vectors (by S) in the columns of A and rows of B are called latent word history vectors for the forward and backward bigrams, respectively. Now, a bigram $w_{ij} = (w_i, w_j)$ $(1 \leq i, j \leq M)$ is represented as a vector of dimension $M \times 1$, where the i^{th} entry of w_{ij} is 1 and the remaining ones are zero. This vector is mapped to a lower dimensional vector $\hat{w}_{ij}$ by:

$$\hat{w}_{ij} = A^T w_{ij} \tag{3}$$

where $\hat{w}_{ij}$ has dimension of $R \times 1$. Similarly, the backward bigram w_{ji} $(1 \leq j, i \leq N)$ is mapped to a lower dimensional vector $\hat{w}_{ji}$ by:

$$\hat{w}_{ji} = B w_{ji} \tag{4}$$

where $\hat{w}_{ji}$ has dimension of $R \times 1$. Note that for a word–morpheme co–occurrence matrix the rows of B would contain latent morpheme vectors.

Since a trigram history consists of two bigram histories, a trigram history vector is obtained by concatenating two bigram vectors. Having generated the features, now we explain the structure of the parametric model and how to train it for language modeling in continuous space.

2.3 Parametric Model Training in Continuous Space

Recalling the necessary inputs to train an acoustic model for speech recognition would be helpful to explain the new language modeling method. The acoustic model training in speech recognition needs three inputs: 1) features (extracted from the speech waveform), 2) transcriptions of the speech waveforms and 3) baseforms, which show the pronunciation of each word in the vocabulary. We propose to model the language model using HMMs. The HMM parameters are estimated in such way that the given set of observations is represented by the model in the "best" way. The "best" can be defined in various ways. One obvious choice is to use Maximum Likelihood (ML) criterion. In ML, we maximize the probability of a given sequence of observations O, belonging to a given class, given the HMM λ of the class, with respect to the parameters of the model λ.

[1]For the generality of the notation, from now on we use "word" instead of "morpheme".

This probability is the total likelihood of the observations and can be expressed mathematically as:

$$L_{tot} = p(O|\lambda) \tag{5}$$

However, there is no known way to analytically solve for the model $\lambda = \{A, B, \pi\}$, which maximize the quantity L_{tot}, where A is the transition probabilities, B is the observation probabilities, and π is the initial state distribution. But we can choose model parameters such that it is locally maximized, using an iterative procedure, like Baum-Welch method (Baum et al., 1970).

The objective function given in Eq. 5 is the same objective function used to estimate the parameters of an HMM based acoustic model. By drawing an analogy between the acoustic model training and language modeling in continuous space, the history vectors are considered as the acoustic observations (feature vectors) and the next word to be predicted is considered as the label the acoustic features belong to, and words with their morphological decompositions can be considered as the lexicon or dictionary. Fig. 1 presents the topology of the model for modeling a word sequence of 3 words. Each word is modeled with a single state left–to–right HMM topology. Using a morphologically rich language (or a character based language like Chinese) to explain how HMMs can be used for language modeling will be helpful. In the figure, let the states be the words and the observations that they emit are the morphemes (or characters in the case of Chinese). The same topology (3 states) can also be used to model a single word, where the first state models the prefixes, the middle state models the stem and the final state models the suffixes. In this case, words are represented by network of morphemes. Each path in a word network represents a segmentation (or "pronunciation") of the word.

The basic idea of the proposed modeling is to create a separate model for each word of the language and use the language model corpus to estimate the parameters of the model. However, one could argue that the basic model could be improved by taking the contexts of the morphemes into account. Instead of building a single HMM for each word, several models could be trained according to the context of the morphemes. These models are called context–dependent morphemes. The most obvious choice is to use both left and right neighbor of a morpheme as context, and creating, what we call tri–morphemes. In principal even if context-dependent morphemes could improve the modeling accuracy, the number of models increase substantially. For a vocabulary size of V, the number of tri–morpheme could be as high as V^3. However, most of the tri–morphemes are either rare or will not be observed in the training data altogether.

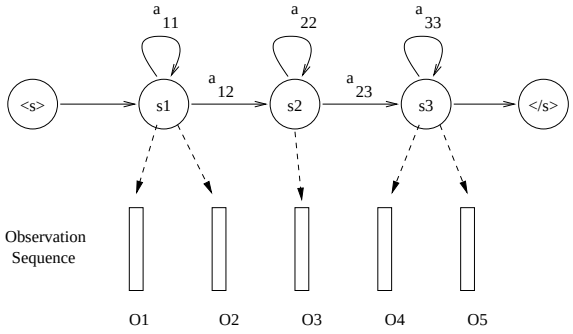

Figure 1: HMM topology for language modeling in continuous space.

Decision tree is one approach that can solve this problem. The main idea is to find similar tri–morphemes and share the parameters between them. The decision tree uses a top-down approach to split the samples, which are in a single cluster at the root of the tree, into smaller clusters by asking questions about the current morpheme and its context. In our case, the questions could be syntactic and/or semantic in nature.

What we hope for is that in the new continuous space there is some form of distance or similarity between histories such that histories not observed in the data for some words are smoothed by similar observed histories.

2.4 Summary of the Continuous Language Model Training and Using it for Decoding

In the upper part of Fig. 2 the language model training steps are shown. The training process starts with the language model training corpus. From the sentences a bigram word co–occurrence matrix is constructed. This is a square matrix where the number of rows (columns) equal to the vocabulary size of the training data. The bigram co–occurrence ma-

trix is decomposed using SVD. The columns of the left–singular matrix obtained from SVD is used to map the bigram word histories into a lower dimensional continuous parameter space. The projected word history vectors are stacked together depending on the size of the n–gram. For example, for trigram modeling two history vectors are stacked together. Even though, we have not done so, at this stage one could cluster the word histories for robust parameter estimation. Now, the feature vectors, their corresponding transcriptions and the lexicon (baseforms) are ready to perform the "acoustic model training". One could use maximum likelihood criterion or any other objective function such as Minimum Phone Error (MPE) training to estimate the language model parameters in the continuous space.

The decoding phase could employ an adaptation step, if one wants to adapt the language model to a different domain, speaker or genre. Then, given a hypothesized sequence of words the decoder extracts the corresponding feature vectors. The feature vectors are used to estimate the likelihood of the word sequence using the HMM parameters. This likelihood is used to compute the probability of the word sequence. Next, we introduce Tied–Mixture Modeling, which is a special HMM structure to robustly estimate model parameters.

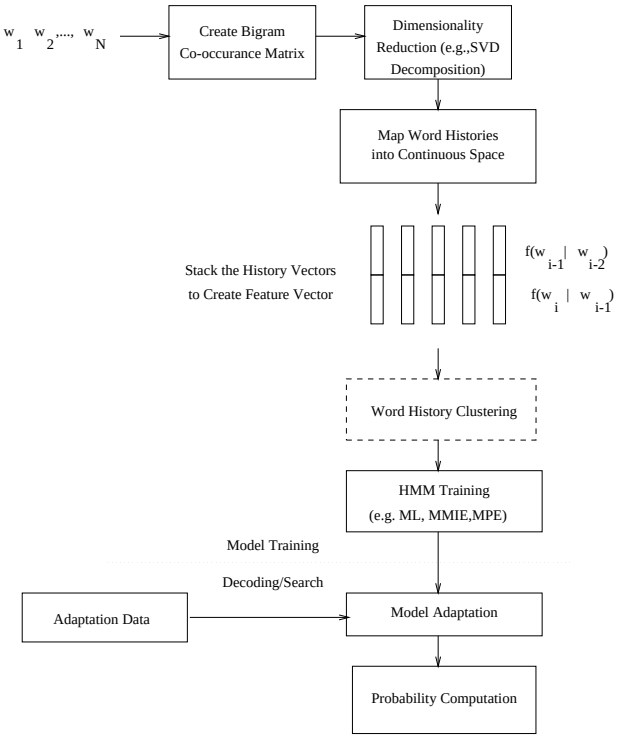

Figure 2: Language Model Training and Adaptation in Continuous Space.

3 Tied–Mixture Modeling

Hidden Markov Models (HMMs) have been extensively used virtually in all aspects of speech and language processing. In speech recognition area continuous-density HMMs have been the standard for modeling speech signals, where several thousand context–dependent states have their own Gaussian density functions to model different speech sounds. Typically, speech data have hundreds of millions of frames, which are sufficient to robustly estimate the model parameters. The amount of data for language modeling is orders of magnitude less than that of the acoustic data in continuous space. Tied–Mixture Hidden Markov Models (TM–HMMs) (Bellegarda & Nahamoo, 1989; Huang & Jack, 1988) have a better decoupling between the number of Gaussians and the number of states compared to continuous density HMMs. The TM–HMM is useful for language modeling because it allows us to choose the number of Gaussian densities and the number of mixture weights independently. Much more data is required to reliably estimate Gaussian densities than to estimate mixture weights.

The evaluation of the observation density functions for TM–HMMs can be time consuming due to the large mixture weight vector and due to the fact that for each frame all Gaussians have to be evaluated. However, there are a number of solutions proposed in the past that significantly reduces the computation (Duchateau et al., 1998).

The function $p(w \mid h)$, defined in a continuous space, represents the conditional probability of the word w given the history h. In general, h contains previous words and additional information (e.g. part-of-speech (POS) tags for the previous words) that may help to the prediction of the next word. Unlike TM–HMMs, using a separate HMM for each word as in the case of Gaussian Mixture Models (GMMs), to represent the probability distribution functions results in the estimation problems for the model parameters since each n–gram does not have hundreds of examples. TM–HMMs use Gaussian mixture probability density functions per

state in which a single set of Gaussians is shared among all states:

$$p(o|w) = \sum_{j}^{J} c_{w,j} \mathcal{N}_j(o, \mu_{w,j}, \Sigma_{w,j}) \qquad (6)$$

where w is the state, $\mathcal{N}_j$ is the jth Gaussian, and o is the observation (i.e. history) vectors. and J is the number of component mixtures in the TM-HMM. In order to avoid zero variance in word mapping into continuous space, all the latent word vectors are added a small amount of white noise.

The TM–HMM topology is given in Fig. 3. Each state models a word and they all share the same set of Gaussian densities. However, each state has a specific set of mixture weights associated with them. This topology can model a word–sequence that consist of three words in them. The TM–HMM estimates the probability of observing the history vectors (h) for a given word w. However, what we need is the posterior probability $p(w \mid h)$ of observing w as the next word given the history, h. This can be obtained using the Bayes rule:

$$p(w|h) = \frac{p(h|w)p(w)}{p(h)} \qquad (7)$$

$$= \frac{p(h|w)p(w)}{\sum_{v=1}^{V} p(h|v)p(v)} \qquad (8)$$

where $p(w)$ is the unigram probability of the word w. The unigram probabilities can also be substituted for more accurate higher order n–gram probabilities. If this n–gram has an order that is equal to or greater than the one used in defining the continuous contexts h, then the TMLM can be viewed as performing a kind of smoothing of the original n–gram model:

$$P_s(w \mid h) = \frac{P(w \mid h)p(h \mid w)}{\sum_{v=1}^{V} P(v \mid h)p(h \mid v)} \qquad (9)$$

where $P_s(w \mid h)$ and $P(w \mid h)$ are the smoothed and original n–grams.

The TM–HMM parameters are estimated through an iterative procedure called the Baum-Welch, or forward-backward, algorithm (Baum et al., 1970). The algorithm locally maximizes the likelihood function via an iterative procedure. This type of training is identical to training continuous density HMMs except the Gaussians are tied across all arcs. For the model estimation equations the readers are referred to (Bellegarda & Nahamoo, 1989; Huang & Jack, 1988).

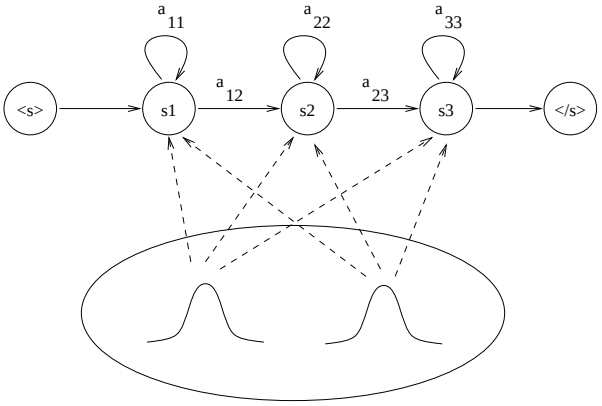

Figure 3: Tied-Mixture HMM topology for language modeling in continuous space. The mixtures are tied across states. Each state represents a word. The TM-HMM is completely defined with the mixture weights, mixture densities and transition probabilities.

Next, we introduce the speech recognition system used for the experiments.

4 Speech Recognition Architecture

The speech recognition experiments are carried out on the Iraqi Arabic side of an English to Iraqi Arabic speech-to-speech translation task. This task covers the military and medical domains. The acoustic data has about 200 hours of conversational speech collected in the context of a DARPA supported speech-to-speech (S2S) translation project (Gao et al., 2006).

The feature vectors for training acoustic models are generated as follows. The speech data is sampled at 16kHz and the feature vectors are computed every 10ms. First, 24-dimensional MFCC features are extracted and appended with the frame energy. The feature vector is then mean and energy normalized. Nine vectors, including the current vector and four vectors from its right and left contexts, are stacked leading to a 216-dimensional parameter space. The feature space is finally reduced from 216 to 40 dimensions using a combination of linear discriminant analysis (LDA), feature space MLLR (fMLLR) and feature space MPE (fMPE) training (Povey et al.,

2005). The baseline speech recognition system used in our experiments is the state–of–the–art and produces a competitive performance.

The phone set consists of 33 graphemes representing speech and silence for acoustic modeling. These graphemes correspond to letters in Arabic plus silence and short pause models. Short vowels are implicitly modeled in the neighboring graphemes. Feature vectors are first aligned, using initial models, to model states. A decision tree is then built for each state using the aligned feature vectors by asking questions about the phonetic context; quinphone questions are used in this case. The resulting tree has about 3K leaves. Each leaf is then modeled using a Gaussian mixture model. These models are first bootstrapped and then refined using three iterations of forward–backward training. The current system has about 75K Gaussians.

The language model training data has 2.8M words with 98K unique words and it includes acoustic model training data as a subset. The morphologically analyzed training data has 58K unique vocabulary items. The pronunciation lexicon consists of the grapheme mappings of these unique words. The mapping to graphemes is one-to-one and there are very few pronunciation variants that are supplied manually mainly for numbers. A statistical trigram language model using Modified Kneser-Ney smoothing (Chen& Goodman, 1996) has been built using the training data, which is referred to as Word-3gr.

For decoding a static decoding graph is compiled by composing the language model, the pronunciation lexicon, the decision tree, and the HMM graphs. This static decoding scheme, which compiles the recognition network off–line before decoding, is widely adopted in speech recognition (Riley et al., 2002). The resulting graph is further optimized using determinization and minimization to achieve a relatively compact structure. Decoding is performed on this graph using a Viterbi beam search.

5 Experimental Results

We used the following TMLM parameters to build the model. The SVD projection size is set to 200 (i.e. $R = 200$) for each bigram history. This results into a trigram history vector of size 400. This vector is further projected down to a 50 dimensional feature space using LDA transform. The total number of Gaussian densities used for the TM–HMM is set to 1024. In order to find the overall relationship between trigram and TMLM probabilities we show the scatter plot of the trigram and TMMT probabilities in Fig. 4. While calculating the TMLM score the TMLM likelihood generated by the model is divided by 40 to balance its dynamic range with that of the n–gram model. Most of the probabilities lie along the diagonal line. However, some trigram probabilities are modulated making TMLM probabilities quite different than the corresponding trigram probabilities. Analysis of TMLM probabilities with respect to the trigram probabilities would be an interesting future research.

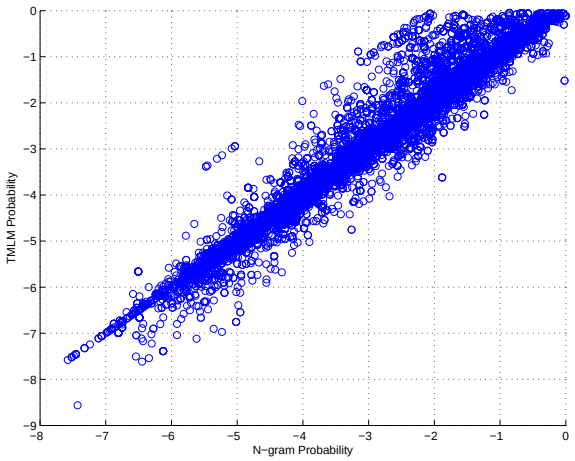

Figure 4: Scatter plot of the n–gram and TMLM probabilities.

We conducted the speech recognition language modeling experiments on 3 testsets: TestA, TestB and TestC. All three test sets are from July'07 official evaluations of the IBM's speech-to-speech translation system by DARPA. TestA consists of sentences spoken out in the field to the IBM's S2S system during live evaluation. TestB contains sentences spoken in an office environment to the live S2S system. Using on-the-spot speakers for TestA and TestB meant to have shorter and clean sentences. Finally TestC contains pre-recorded sentences with much more hesitations and more casual conversations compared to the other two testsets. TestA, TestB and TestC have 309, 320 and 561 sentences, respectively.

LM	TestA	TestB	TestC	All
Word-3gr	18.7	18.6	38.9	32.9
TMLM	18.8	18.9	38.2	32.5
TMLM + Word-3gr	17.6	18.0	37.4	31.9

Table 1: Speech Recognition Language Model Rescoring Results.

In order to evaluate the performance of the TMLM, a lattice with a low oracle error rate was generated by a Viterbi decoder using the word trigram model (Word-3gr) model. From the lattice at most 30 (N=30) sentences are extracted for each utterance to form an N-best list. The N–best error rate for the combined test set (All) is 22.7%. The N–best size is limited (it is not in the hundreds), simply because of faster experiment turn-around. These utterances are rescored using TMLM. The results are presented in Table 1. The first two rows in the table show the baseline numbers for the word trigram (Word–3gr) model. TestA has a WER of 18.7% similar to that of TestB (18.6%). The WER for TestC is relatively high (38.9%), because, as explained above, TestC contains causal conversation with hesitations and repairs, and speakers do not necessarily stick to the domain. Moreover, when users are speaking to a device, as in the case of TestA and TestB, they use clear and shorter sentences, which are easier to recognize. The TMLM does not provide improvements for TestA and TestB but it improves the WER by 0.7% for TestC. The combined overall result is a 0.4% improvement over baseline. This improvement is not statistically significant. However, interpolating TMLM with Word-3gr improves the WER to 31.9%, which is 1.0% better than that of the Word-3gr. Standard p-test (Matched Pairs Sentence-Segment Word Error test available in standard SCLITEs statistical system comparison program from NIST) shows that this improvement is significant at $p < 0.05$ level. The interpolation weights are set equally to 0.5 for each LM.

6 Discussions

Despite limited but encouraging experimental results, we believe that the proposed perspective is a radical departure from the traditional n–gram based language modeling methods. The new perspective opens up a number of avenues which are impossible to explore in one paper.

We realize that there are a number of outstanding issues with the proposed perspective that require a closer look. We make a number of decisions to build a language model within this perspective. The decisions are sometimes ad hoc. The decisions are made in order to build a working system and are by no means the best decisions. In fact, it is quite likely that a different set of decisions may result into a better system. Using a word–morpheme co–occurrence matrix instead of a morpheme–morpheme co–occurrence matrix is one such decision. Another one is the clustering/tying of the rarely observed events to achieve robust parameter estimation both for the SVD and TMLM parameter estimation. We also use a trivial decision tree to build the models where there were no context questions. Clustering morphemes with respect to their syntactic and semantic context is another area which should be explored. In fact, we are in the process of building these models. Once we have realized the full potential of the baseline maximum likelihood TMLM, then we will investigate the discriminative training methods such as MPE (Povey & Woodland, 2002) to further improve the language model performance and adaptation to new domains using MLLR (Legetter & Woodland, 1995).

We also realize that different problems such as segmentation (e.g. Chinese) of words or morphological decomposition of words into morphemes can be addressed within the proposed perspective.

7 Conclusions

We presented our progress in improving continuous-space language modeling. We proposed the Tied-Mixture Language Model (TMLM), which allows for robust parameter estimation through the use of tying and improves on the previously presented GMLM. The new formulation lets us train a parametric language model using off–the–shelf acoustic model training tools. Our initial experimental results validated the proposed approach with encouraging results.

References

M.Afify, O. Siohan and R. Sarikaya. 2007. *Gaussian Mixture Language Models for Speech Recognition*, ICASSP, Honolulu, Hawaii.

C.J. Legetter and P.C. Woodland. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, *Computer Speech and Language*, vol.9, pp. 171-185.

J. Bellegarda. 2000. Large Vocabulary Speech Recognition with Multispan Language Models, *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 76-84.

H. Schwenk, and J.L. Gauvain. 2003. Using Continuous Space Language Models for Conversational Telephony Speech Recognition, *IEEE Workshop on Spontaneous Speech Processing and Recognition*, Tokyo, Japan.

J. Duchateau, K. Demuynck, D.V. Compernolle and P. Wambacq. 1998. Improved Parameter Tying for Efficient Acoustic Model Evaluation in Large Vocabulary Continuous Speech Recognition. *Proc. of ICSLP*, Sydney, Australia.

Y. Gao, L. Gu, B. Zhou, R. Sarikaya, H.-K. Kuo. A.-V.I. Rosti, M. Afify, W. Zhu. 2006. IBM MASTOR: Multilingual Automatic Speech-to-Speech Translator. *Proc. of ICASSP*, Toulouse, France.

S. Chen, J. Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling, *ACL*, Santa Cruz, CA.

J. Bellagarda and D. Nahamoo. 1989. Tied mixture continuous parameter models for large vocabulary isolated speech recognition, *Proc. of ICASSP*, pp. 13-16.

X.D. Huang and M.A. Jack. 1988. Hidden Markov Modelling of Speech Based on a Semicontinuous Model, *Electronic Letters*, 24(1), pp. 6-7, 1988.

D. Povey and P.C. Woodland. 2002. Minimum phone error and I-smoothing for improved discriminative training, *Proc. of ICASSP*, pp. 105–108, Orlando, Florida.

D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, G. Zweig. 2005. fMPE: Discriminatively Trained Features for Speech Recognition, *Proc. of ICASSP*, pp. 961–964, Philadelphia, PA.

C. Chelba and F. Jelinek. 2000. Structured language modeling, *Computer Speech and Language*, 14(4), 283–332, 2000.

H. Erdogan, R. Sarikaya, S.F. Chen, Y. Gao and M. Picheny. 2005. Using Semantic Analysis to Improve Speech Recognition Performance, *Computer Speech & Language Journal*, vol. 19(3), pp: 321–343.

B. Roark, M. Saraclar, M. Collins. 2007. Using Semantic Analysis to Improve Speech Recognition Performance, *Computer Speech & Language*, vol. 21(2), pp: 373–392.

M. Riley, E. Bocchieri, A. Ljolje and M. Saraclar. 2007. The AT&T 1x real-time Switchboard speech-to-text system, *NIST RT02 Workshop*, Vienna, Virginia.

Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, vol. 3, 11371155.

S. Deerwester, Susan Dumais, G. W. Furnas, T. K. Landauer, R. Harshman. 1990. Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, 41 (6): 391–407.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. 1970. A Maximization Techniques Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains, *The Annals of Mathematical Statistics*, 41(1):164–171.