LANGUAGE MODELING USING INDEPENDENT COMPONENT ANALYSIS FOR AUTOMATIC SPEECH RECOGNITION

Raghunandan S. Kumaran, Karthik Narayanan and John N. Gowdy

Department of Electrical and Computer Engineering, Clemson University,
Clemson, SC 29634 USA
{ksampat, knaraya, jgowdy}@clemson.edu

ABSTRACT

Conventional statistical language models such as N-grams are inadequate to model long distance dependencies in natural language. In this paper we propose a novel statistical language model to capture topic related long range dependencies. Humans have the inherent ability to identify long range dependencies in natural language. Given a set of related words humans can easily identify the context in which the set of words is occurring. It has been shown by many researchers that Independent Component Analysis (ICA) captures these kind of dependencies better than any other formulation. Furthermore, ICA provides a topic decomposition that can be easily interpreted by humans compared to other models. This paper describes the development of a language model using ICA. The topic model is combined with a standard N-gram to produce the language model. The perplexity results obtained show that this language model is a viable language model for speech recognition purposes.

1. INTRODUCTION

The goal of statistical language modeling is to assign probabilities to a sequence of words. The most prominent use of language models is in Automatic Speech Recognition (ASR), where the language model assigns a *priori* probability to help differentiate words that have similar acoustical properties. The most popular language models in use are the N-grams. Although they are effective for some applications, their predictive power is limited. Usually N is of the order of 2 or 3, which greatly restricts the predictive power of the N-grams. Higher order N-grams (of the order of N = 6,7,...) have been tried, but they have been found to be unreliable, the main reason for this being data sparseness. Even higher order N-grams cannot capture the long range dependencies of natural language, which humans can easily identify. Several attempts have been made to overcome this limitation. The earliest attempts in this regard were made in the form of variable length word-category based N-grams [1]. In this method an attempt was made to classify words into categories and then use these for language modeling. Cache models [2, 3, 4] try to increase the probability of the words in the history by allowing the probability to decay exponentially according to the distance. Trigger-based models [5, 6] work on the principle of word trigger pairs in which the probability of a particular word increases if its trigger pair is in the history. A more recent approach is the Latent Semantic Analysis (LSA) language model [7] which uses semantic relationships to model the language. Grammar based techniques have also been attempted. These try to exploit syntactical regularities to model the long range dependencies. Prominent approaches of this kind are described in [8, 9]. A more recent approach [10] tries to use syntactic information in the LSA type formulation to model the long range dependencies.

Our approach here is more closely related to the LSA formulation [7] in that we try to use a semantic relationship between words and documents. Mathematically the relationship can be written as

$$P(w|h) = \sum_t P(w|t)P(t|h) \qquad (1)$$

where $P(w|t)$ is the topic specific word probability and $P(t|h)$ is the mixing factor that depends on the history h, and t is the variable that refers to different topics. This latent variable t is produced by ICA.

ICA is a method of representing a set of multivariate observations as a linear combination of unknown latent variables that are statistically independent. ICA was originally developed in the context of Blind Source Separation (BSS) but has lately been found useful in text document analysis. The first attempts at using ICA for text document analysis were performed by [11, 12] in the context of information retrieval where it was very successful. The latent variables in the case of ICA are the topics and these can be regarded as probability distributions on the universe of terms. The resulting model ignores the syntactic information and also the word order as do similar models [13, 7]. This kind of modeling implies a single word-topic relation. The local word distributions are handled at a different stage when the ICA model is combined with N-grams.

2. INDEPENDENT COMPONENT ANALYSIS

In this section we give a brief outline of Independent Component Analysis. The classic ICA model can be expressed as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \qquad (2)$$

where $\mathbf{x} = \{x_1, x_2, x_3, .., x_n\}^T$ is the vector of observed random variables. The vector of the independent latent variables is denoted by $\mathbf{s} = \{s_1, s_2, s_3, .., s_n\}^T$ and $\mathbf{A}$ is an unknown constant matrix called the *mixing matrix*. If we denote the columns of matrix $\mathbf{A}$ by a_i the model can be written as

$$\mathbf{x} = \sum_{i=1}^{n} a_i s_i. \qquad (3)$$

The goal in ICA is to learn the decomposition in Eq. 2, i.e., $\mathbf{A}$ and $\mathbf{s}$, in an unsupervised manner. That is we only observe

x and want to estimate both **A** and **s**. The basic assumption in ICA is that the latent variables s_i are statistically independent. Also, there are three properties of ICA that must be taken into account when considering the results. First, one cannot determine the variances of the independent components. Second, one cannot determine the order of the components, and third, the independent components must be non-gaussian. At most, one the independent components can be gaussian. Once the mixing matrix is estimated, the independent components can be obtained by

$$\mathbf{s} = \mathbf{W}\mathbf{x}. \tag{4}$$

where $W = A^{-1}$, is called the *scattering matrix*. The principle of ICA is 'non-gaussian is independent'. The Central Limit Theorem of probability theory states that a linear combination of independent random variables tends toward a gaussian distribution under certain conditions. Hence, the sum of many random variables will be least gaussian if only one of the random variables contributes significantly towards the sum. Using this approach we can estimate each one of the independent components. By estimating the non-gaussanity in all the n-dimensions, we can estimate all the n independent components.

3. STATISTICAL LANGUAGE MODELING USING ICA

In this section we show how we can utilize the ICA framework in language modeling using the concept of ICA explained above.

3.1 ICA Framework

The ICA framework is applied to raw text consisting of M documents and spanning a vocabulary V of N words. In our case N = 10,000 and M = 9,371. The documents consists of a few short paragraphs each containing typically less than 10 sentences. The important point here is that all the text in a document relates to the same subject and hence is suitable for a 'semantic analysis' [7] kind of test to learn the semantic relationships between the words and documents.

First, we need to pre-process the term document matrix to produce a feature matrix **W** that can emphasize the importance of the words in a document. For this purpose we follow the same feature matrix representation as in [7]. The cell entries of **W** are obtained as follows ,

$$w_{i,k} = (1-\varepsilon_i)\frac{c_{i,k}}{n_k} \tag{5}$$

and

$$\varepsilon_i = -\frac{1}{\log K}\sum_{k=1}^{K}\frac{c_{i,k}}{t_i}\log\frac{c_{i,k}}{t_i} \tag{6}$$

where $c_{i,k}$ is the number of times words w_i occurs in document d_k, n_k is the total number of words present in document d_k, $t_i = \sum_k c_{i,k}$ and ε_i is the normalized entropy of $w_i$in the corpus.

The next step in the process is to apply ICA to this word document matrix. The ICA model that we are adopting here is a combination of ICA and SVD decompositions giving

$$W_{txd} = T_{txk}A_{kxk}S_{kxd} \tag{7}$$

where the matrix T holds the term eigenvectors, A is the ICA mixing matrix and S holds the separated documents or the 'topics'. SVD decomposition is given by

$$W = TLD^T \tag{8}$$

where matrix L contains the singular values. Using Eq. 8 and by matrix inversion of A, the independent component matrix S is found by, $S = A^{-1}LD^T$ This method is similar to [12]. The value chosen for k should reflect the number of different topics that are present in the corpus. Any new document can be projected onto this space to determine which topic it belongs to. Consider a new document containing words $w_{i_1}, w_{i_2}, ..., w_{i_n}$. The preprocessing, as described previously, is performed to obtain the weighted document vector **d**. Projecting this document vector onto the space, we obtain **v** as

$$\mathbf{v} = A^{-1}T^T\mathbf{d}. \tag{9}$$

The elements of vector **v** gives us a weighted estimate of how many topics this document belongs to. If v_i is the largest component of vector **v**, we can conclude that this document belongs to the topic s_i (i^{th} independent component). Now if we are given 2 documents, then they can be compared for semantic similarity by calculating the cosine between their projections.

3.2 Language modeling using ICA

We have seen in the previous section how we can obtain semantic relationships between documents using ICA. Now we will see how this ability of ICA can be used to develop a language model which assigns higher probability to the words that are semantically close to the current history and lower probabilities to other words. As discussed in the previous section, any document vector can be projected onto the ICA space. Therefore, consider the current history of words $w_{q-1}, w_{q-2}, ..., w_1$ as a 'pseudo document' and project this pseudo document onto the ICA space. This projection will give us the current topic. Next we need to determine how close the current word w_q is to this topic. If the word is very close to the current topic, then we assign it a high probability and if it is not close, then assign it a low probability. For this purpose we need to define a metric that can determine this semantic closeness. We can use a similar metric as used in [7]. From Eq. 7, T contains the term vectors. We need to determine how close the term vector corresponding to the given word is to the topic. That is, we need to find out how close the vector t_iA, corresponding to word w_i, is to $A\tilde{s}_j$. A simple distance measure available is the dot product. Thus,

$$D(w_q, H_{1,q-1}) = (t_qA).(A\tilde{s}_{q-1}) \tag{10}$$

where $H_{1,q-1}$ is the history, $\tilde{s}_{q-1}$ is the projection of the history on to the ICA space and t_q is the term vector corresponding to word w_q.

The normalized version of Eq.10 is

$$D(w_q, H_{1,q-1}) = \left[\frac{t_qA}{||t_qA||}\right]\left[\frac{A\tilde{s}_{q-1}}{||A\tilde{s}_{q-1}||}\right]^T \tag{11}$$

This distance metric can be used to generate a probability value by converting it to a probability mass function. This is done by making the total probability over all the words to be equal to 1, such that the farther away a word is from the current topic of the history, the lower its probability will be. Therefore, the ICA probability can be obtained by

$$P^{ica}(w_q|H_{1,q-1}) = \frac{D(w_q,H_{1,q-1})}{\sum_{w_i \in V} D(w_i,H_{1,q-1})}. \qquad (12)$$

This ICA probability measure can capture the long distance relationships of the words, but cannot model the local word distributions. N-grams do the exact opposite. They can capture the local word distributions well because of maximum likelihood estimation from the training corpus and various smoothing techniques, but not the long distance relationships. Logic suggests combining the two to obtain the best performance. To combine the two models we have used an approach similar to the one in [7]. We use

$$P(w_q|H_{q-1}) = \frac{P(w_q,H^{(l)}_{1,q-1}|H^{(n)}_{q-n+1,q-1})}{\sum_{w_i \in V} P(w_i,H^{(l)}_{1,q-1}|H^{(n)}_{q-n+1,q-1})} \qquad (13)$$

where $H^{(l)}_{1,q-1}$ is due to the history of ICA model and $H^{(n)}_{q-n+1,q-1}$ is the history of the N-gram. The final model is given by

$$P(w_q|H_{q-1}) = \frac{P^{(ica)}(w_q,H_{1,q-1})P(w_q|w_{q-1,\dots,q-n+1})}{\sum_{w_i \in V} P^{(ica)}(w_i,H_{1,q-1})P(w_i|w_{q-1,\dots,q-n+1})} \qquad (14)$$

4. EXPERIMENTS AND RESULTS

In this section we describe our experimental setup and the results that were obtained. An analysis of the results obtained is also presented in this section.

The performance of any language model is measured in terms of *perplexity*, which is a measure of how well the model predicts the occurrence of words in a given (unseen) text. The assumption is that the test text is from the same domain as the training data. The perplexity of a language model **X** is given by

$$PP(\mathbf{X}) = -\frac{1}{N}\sum_{i=1}^{N} \log P(w_i|H_{q-1}) \qquad (15)$$

where N is the total number of words in the test set. Perplexity also indicates the average branching produced by the language model **X**. Thus, perplexity indicates how well the model performs for a speech recognition task. Generally, the relation between word error rate of a speech recognizer and the perplexity is linear, that is, lower perplexity usually translates to lower word error rates. We have implemented our model on the Wall Street Journal database from the BLLIP corpus, which is a collection of news stories from the Wall Street Journal for the years 1987, 1988 and 1989.

The WSJ corpus consists of about 94,000 documents and 43 million words. For this experiment we used the data from year 1987. This data consists of about 41,275 documents and about 20 million words. We have used the 9,371 largest documents among these as a trade-off between data sparseness and computational complexity. The vocabulary consists of 10,000 most frequently occurring words in the dataset. Note that in contrast to others [10], we made no attempt to remove function words such as 'the', 'and', 'to', as in [7]. This was done keeping in mind that removing the function words may affect the language model, even though these kind of words do not carry any semantic information. The dataset was split into a training set of about 4.7 million words and a test set of about 250,000 words. All numerical strings were mapped to the same number '9' as in [10].

Our word-document matrix is of size 10,000 x 9,371. ICA was implemented on Matlab using the FastICA toolkit [14]. The word-document matrix has about 99% sparseness. Hence, the word-document matrix was stored in sparse matrix format in Matlab resulting in a significant saving of memory. The SVD decomposition of Eq. [8] was implemented using the sparse matrix toolbox of Matlab. We have used both Bi-gram and Tri-gram as the N-gram in our model. These were implemented using the CMU-Cambridge toolkit [15]. The perplexity was obtained for k = 50, 100 and 200 for each case, i.e. ICA + Bi-gram and ICA + Tri-gram.

Baseline Bi-gram	246.3

No of *Topics*	Perplexity	% reduction from Bi-gram
50	152.3	38.1%
100	143.8	41.54%
200	134.2	45.45%

Figure 1: *Perplexity results for ICA + Bi-gram*

The results of combining the ICA model with a Bi-gram is given in Figure. 1. From the results we can see that ICA model has been able to reduce the perplexity from the Bi-gram value. The maximum reduction in perplexity achieved was 45.45%. Also as the number of topics increased, the perplexity reduces. This means that we are yet to reach the actual number of topics that are in the training set. Ideally, k should be equal to the actual number of topics that are in the data set.

Baseline Tri-gram	166.1

No. of *Topics*	Perplexity	% reduction from Tri-gram
50	115.73	30.28%
100	106.9	35.6%
200	102.78	38.08%

Figure 2: *Perplexity results for ICA + Tri-gram*

The results of combining the ICA model with a Tri-gram are given in Figure. 2. From the results we can see that ICA model has been able to reduce the perplexity from the Tri-gram value. The maximum reduction in perplexity achieved was 38.08%. Also, the perplexity is less than the case of ICA

+ Bi-gram. This is expected as the baseline perplexity of a Tri-gram is far less than that of Bi-gram. But the reduction in perplexity from the baseline is not as much as in the Bi-gram case. As in the Bi-gram case, the perplexity reduces as the number of topics increases. Comparing the results with that of the Bi-gram case, we can say that ICA + Tri-gram is a better model. Also, implementing a Tri-gram is not significantly more difficult than a Bi-gram. Thus ICA + Tri-gram is the better model among the two.

5. ANALYSIS OF RESULTS

The results in the previous section show that the ICA based model is a viable model for language modeling. The ICA + Trigram model has performed better than the ICA + Bi-gram model. This is expected, since this model is similar to LSA model [7] where similar results have been presented. A true comparison of the two models, however cannot be made since the results in [7] have been obtained on a different test and training set. However based on the results reported in [11], where a comparison between LSA and ICA based models was done for Information Retrieval purposes, we can say that ICA based model will definitely preform better than the LSA model for language modeling. However at this point in time we do not have any experimental evidence to confirm this. [7] has been able to improve upon the baseline perplexities by applying different smoothing techniques. This suggests that the same could be done here to improve the perplexity of the ICA based language model. Also implementing the ICA based is very when compared to something like EM based model where the EM algorithm has to be implemented. In this sense the ICA model is advantageous when compared to the EM based model of [13].

6. CONCLUSION AND FUTURE WORK

We have successfully demonstrated a method of applying Independent Component Analysis for statistical language modeling. A topic model was developed using ICA. The topic model was then combined with a standard N-gram to obtain the language model. The perplexity results obtained compare favorably with other other models such as the LSA model [7] and EM based model [13] and is very encouraging. Our immediate plan for the future includes applying better smoothing techniques for the language model. Word clustering techniques could also be applied to further improve the perplexity of the model. Techniques to incorporate syntactic information will also be investigated in the future as well as testing this model on a real time speech recognition task.

REFERENCES

[1] T. Niesler and P. Woodland, “A variable-length category-based n-gram language model,” in *Proc. ICASSP ’96*, Atlanta, GA, 1996, pp. 164–167.

[2] R. Kuhn and R. De Mori, “A cache-based natural language model for speech recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 6, pp. 570–583, 1990.

[3] P. Clarkson and A. Robinson, “Language model adaptation using mixtures and an exponentially decaying cache,” in *Proc. ICASSP ’97*, Munich, Germany, 1997, vol. 2, p. 799, IEEE Computer Society.

[4] R. Iyer and M. Ostendorf, “Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models,” in *Proc. ICSLP ’96*, Philadelphia, PA, 1996, vol. 1, pp. 236–239.

[5] R. Rosenfeld, “A maximum entropy approach to adaptive statistical language modeling,” in *R. Rosenfeld, A Maximum Entropy Approach to Adaptive Statistical Language Modeling, Computer, Speech and Language, vol. 10, pp. 187– 228, 1996. Longer version: Carnegie Mellon Tech. Rep. CMU-CS-94-138.*, 1996.

[6] R. Lau, R. Rosenfeld, and S. Roukos, “Trigger-based language models: A maximum entropy approach.,” in *Proc. ICASSP ’93*, Minnesota, MN., 1993, pp. 45–48.

[7] J. Bellegarda, “A multispan language modeling framework for large vocabulary speech recognition,” in *IEEE Trans. Speech Audio Processing*, 1998, vol. 6, pp. 456–467.

[8] C. Chelba and F. Jelinek, “Exploiting syntactic structure for language modeling,” in *Proc. of the 36th conference on Association for Computational Linguistics*, Montreal, Quebec, Canada, 1998, pp. 225–231, Association for Computational Linguistics.

[9] W. Wang, A. Stolcke, and M. Harper, “The use of a linguistically motivated language model in conversational speech recognition,” in *Proc. ICASSP ’04*, Montreal, Quebec, Canada, 2004.

[10] D. Kanejiya, A. Kumar, and S. Prasad, “Statistical language modeling using syntactically enhanced lsa,” in *WSLP-2003*, 2003, pp. 93–100.

[11] C. L. Isbell, Jr. and P. Viola, “Restructuring sparse high dimensional data for effective retrieval,” in *Proc. of the 1998 Conference on Advances in Neural Information Processing Systems II.* 1998, pp. 480–486, MIT Press.

[12] T. Kolenda and L. K. Hansen, “Independent components in text,” in *Advances in Independent Component Analysis*, M.Girolami, Ed., chapter 13, pp. 235–256. Springer-Verlag, 2000.

[13] D. Gildea and T. Hofmann, “Topic-based language models using em,” in *Proc. of the 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, 1999.

[14] A. Hyvrinen and E. Oja, “Independent component analysis: Algorithms and applications,” in *Neural Networks*, 2000, pp. 411–430.

[15] P. Clarkson and R. Rosenfeld, “Statistical language modeling using the CMU–cambridge toolkit,” in *Proc. Eurospeech ’97*, Rhodes, Greece, 1997, pp. 2707–2710.