Dimension-Reduced Estimation of Word Co-occurrence Probability

Kilyoun Kim and **Key-Sun Choi**
Computer Science Dept., AITrc, KORTERM
Korea Advanced Institute of Science & Technology (KAIST)
Kusong-Dong, Yusong-Gu Taejon, 305-701 Republic of Korea
gykim@world.kaist.ac.kr and kschoi@world.kaist.ac.kr

Abstract

We investigate a novel approach to solve the problem of sparse data through dimension reduction. Linear algebraic technique called LSA/SVD is used to find co-relationships of sparse words. Three variant estimation methods are suggested and they are evaluated for estimating unseen noun-verb co-occurrence probability. The model shows possibility to be alternative probability smoothing method.

1 Introduction

One of the most suffering difficulties in statistical language processing is so-called data sparseness problem. No matter how large the training set is, a substantial portion of the data is unseen. For them, the Maximum Likelihood Estimation (MLE) probabilities are zero and these zeros give us bad result all through the statistical process.

We are interested in $P(x, y)$ and the prediction task $P(y|x)$, that is a bigram language modeling of word co-occurrences. $P(y|x)$ is the conditional probability that a pair has second element $y \in Y$ given that its first element is $x \in \mathcal{X}$. In other words, $P(y|x)$ can be regarded as a measure of relationship between word x and y. For example, for a given object $x = beer$, a verb $y = drink$ is more related than a verb $y = eat$, $p(drink|beer) >> p(eat|beer)$. Many features can be used to predict a relationship between two words, but we assume here that the only information we have are the frequencies.

To overcome the difficulty of sparse data, a smoothing technique like Good-Turing method is widely used. Estimator combining approaches such as linear interpolation and Katz's back-off method are popular also(Katz, 1987). They use unigram probability $P(y)$ to estimate bigram probability $P(y|x)$ for unseen data pair, disregarding the relationship between two words. If unseen bigrams are made up of unigrams of the same frequency, the methods give them the same probability, causing a problem to estimate accurate probability.

In addition to the classical methods, similarity-based schemes are successfully applied to data sparseness problem. The nearest-neighbors similarity-based method uses a set of k most similar words x' to estimate conditional probability $P(y|x)$, being said to perform almost 40% better than back-off (Dagan et al., 1999). They use various distributional similarity measures to find similarity between words such as KL-divergence or JS-divergence (Lee, 1999). For the sparse word, however, the distribution $P(y|x)$ itself is sparse and it is difficult to find correct similarity between words, since the only means for measuring word similarity is the frequency. The more sparse the distribution of word is, the more difficult finding acceptable similarities between words.

In this paper, we investigate a novel approach to solve the problem of sparse data by capturing their latent relationships with only frequency information. Through reducing dimension by linear algebraic technique LSA/SVD[1], we can eliminate zero values in

[1]LSA - Latent Semantic Analysis, SVD - Singular

$p(y|x)$ as well as we can capture relationships between words. We believe that the dimension-reduced estimation model can be alternative probability smoothing method.

The model consists of three parts: making a conditional probability matrix, projecting the matrix into lower space, and estimating probabilities on reduced space. In the third part, three variant estimating methods are suggested and they are compared with Katz's back-off method and simplified nearest-neighbor similarity-based method. We evaluated the methods in a pseudo wrod sense disambiguation task. and made promising result. Futher evaluation is needed on more realistic task, though.

The optimal dimension size of subspace is also investigated, showing the best result between $90 - 200$, about 10% of the original dimension size. Finally, we show that the model does not degrade performance as the sparseness increases.

2 Dimension-Reduced Model

Dimension-reduced model uses linear algebraic technique called LSA/SVD, which projects a matrix into reduced space.

First of all, to apply linear algebraic technique, we need to represent conditional probability $P(y|x)$ as a matrix form (Section 2.1). After that, we project the matrix into lower dimension subspace through SVD. We will show how the resultant space represents relationship between the given word x and the predicting word y well (Section 2.2). At last, we suggest three probability estimation methods on reduced space (Section 2.3).

2.1 Conditional Probability Matrix

Any discrete conditional probability distributions can be represented by a matrix form. For a distribution $p(y|x)$, given words $x \in \mathcal{X}$ make up row entries and predicting words $y \in \mathcal{Y}$ make up column entries. Each element of matrix has estimated conditional probability value of two words $p(y|x)$. We define conditional probability matrix and the row, column vectors:

$$A_{m\times n} = [a_{ij}] = [P(y_j|x_i)] \quad (1)$$

$$\vec{x_i} = [P(y_1|x_i), \cdots, P(y_n|x_i)] \quad (2)$$

$$\vec{y_j} = [P(y_j|x_1), \cdots, P(y_j|x_m)]$$

where $m = |\mathcal{X}|$, $n = |\mathcal{Y}|$ and $1 \leq i \leq m$, $1 \leq j \leq n$. For example, if we use MLE estimator, $a_{ij} = P_{MLE}(y_j|x_i) = \frac{frq(x_i,y_j)}{frq(x_i)}$.

In the table, the noun "coffee" and "beer" does not co-occur with the same verb and it is difficult to find similarity between them in this space. To find their latent relationship, we can project each row and column vector into lower dimension space through latent semantic anaylsis.

Table 1 shows an example of MLE estimating matrix. The task is predicting the main verb with given object, that is estimating noun and verb co-occurrence probability $p(v|n)$ where $n \in N, v \in V$. Note that each noun can be regarded as a point or a vector in multi-dimensional space of which a dimension size equal to $|V|$.

Table 1: An Example of Conditional Probability Matrix

$N\|V$	*swig*	*sip*	*drink*	*devour*	*eat*	*swallow*
beer	0.33	0	0.33	0.33	0	0
whiskey	0	0.5	0.5	0	0	0
coffee	0	1	0	0	0	0
bread	0	0	0	0.33	0.33	0.33
sugar	0	0	0	0	0.5	0.5

2.2 Projection - Latent Semantic Analysis

Latent Semantic Analysis (LSA) is known as a theory for extracting and representing the contextual-usage meaning of words. LSA uses singular value decomposition (SVD). It has been widely used in information retrieval task as a variant of the vector space model(Deerwester et al., 1990)(Dumais et al., 1997).

Given the conditional probability matrix A and $rank(A) = r$, the SVD of A and the rank-

Value Decomposition

k approximation matrix A_k is defined as

$$A = U\Sigma V^T = \sum_{i=1}^{n} u_i \cdot \sigma_i \cdot v_i^T \quad (3)$$

$$A_k = U_k \Sigma_k V_k^T = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T \quad (4)$$

where U and V contains left and right singular vectors of A, respectively, and the $\Sigma = diag(\sigma_1, \cdots, \sigma_n)$ is the diagonal matrix of singular values of A. Truncated SVD A_k, which is constructed from the k-largest signular triples of A, is the closest rank-k matrix to A [2]. The left singular vector $\vec{u_i}$ and the right singular vector $\vec{v_i}$ corresponds to the row vector $\vec{x_i}$ and the column vector $\vec{y_i}$, respectively. By taking k elements of $\vec{u_i}$ and $\vec{v_i}$, each given word x and predicting word y of $P(y|x)$ is represented as a vector in the reduced k-space.

Figure 1 is an example of SVD on the noun-verb conditional probability matrix of Table 1. In Figure 1-D, both noun x and verb y are represented by a vector in two dimensional space. Nouns which occur with similar verbs are grouped each other even if they never co-occur with the same verb (dim1: beverages, dim2: foods). For example, noun "coffee" and verb "beer" do not co-occur with the same verb in the original matrix (Table 1); however they are near in two dimensional space when measured with a cosine distance. This means that unseen word pairs (x, y) which do not co-occur in the training data may none the less be near in reduced k-space. This derived representation which captures word(x)-word(y) associations is used for estimating probabilities of unseen data.

2.3 Estimating Probabilities on Reduced Space

Until now, we constructed word co-occurrence probability matrix and projected the matrix into lower dimension space. Now, we suggest three variant probability estimation methods in dimension-reduced space. First is estimating $p(y|x)$ by computing distance between given word x and predicting word y in reduced space. Second, we can use rank-k approximation matrix. Third, the state-of-the-art similarity-based methods can be merged to our dimension-reduced model. Because the first two methods are not based on statistical theory, it should be explored

2.3.1 Method 1: *Distance-based method*

Through LSA, the matrix A is factored into the product of three matrices as in Equation 3, and $\vec{u_i}$ and $\vec{v_j}$ are considered as the row vector $\vec{x_i}$ and the column vector $\vec{y_j}$ in k-dimension subspace respectively. Figure 1-D shows 2-dimensional plot of resultant U_2, V_2 matrix. The distance-based method use normalized distance between $\vec{u_i}$ and $\vec{v_j}$ for estimating probability $P(y_j|x_i)$:

$$P(y_j|x_i) = \frac{1}{Z_k} \cdot \frac{D_k(\vec{u_i}, \vec{v_j}) + 1}{2}$$

$$D_k(\vec{u_i}, \vec{v_j}) = \frac{\sum_{t=1}^{k} u_i(t) v_j^T(t)}{\sqrt{\sum_{t=1}^{k} u_i(t)^2} \sqrt{\sum_{t=1}^{k} v_j(t)^2}}, \quad (5)$$

where Z_k is normalizing factor and D_k is a cosine distnace in k-dimensional space.

2.3.2 Method 2: *Rank-k approximation matrix method*

In LSA, we can create a rank-k approximation matrix A_k to the matrix A by setting all but the k largest singular values of A equal to zero (Equation 4). In this method, we consider each element of a rank-k approximation matrix A_k as probability distribution of $p(y|x)$ (Figure 1-C). To satisfy the requirements $\Sigma p(y|x) = 1$ and $p(y|x) \geq 0$, we use the following normalizing equation:

$$P(y_j|x_i) = \frac{1}{Z_k}\left[A_k(i,j) - \min_v A_k(i,j) + \delta\right],$$
$$Z(x) = \sum_v \left[A_k(i,j) - \min_v A_k(i,j) + \delta\right] \quad (6)$$

where Z(n) is normalizing factor and δ is a smoothing constant.

[2] In other words, the projection into the reduced space is chosen such that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences. One can prove that A_k is the best approximation to A for any unitray invariant norm(Michael W. Berry and Jessup, 1999)

A. Conditional Probability Estimation Matrix by naive frequency

$$A = [P(v|n)] = \left(\begin{array}{c|cccccc} & swig & sip & drink & devour & eat & swallow \\ \hline beer & 0.3333 & 0 & 0.3333 & 0.3333 & 0 & 0 \\ whiskey & 0 & 0.5000 & 0.5000 & 0 & 0 & 0 \\ coffee & 0 & 1.0000 & 0 & 0 & 0 & 0 \\ bread & 0 & 0 & 0 & 0.3333 & 0.3333 & 0.3333 \\ sugar & 0 & 0 & 0 & 0 & 0.5000 & 0.5000 \end{array}\right)$$

B. Singular Vector Decomposition

$$A = U\Sigma V^T =$$

$$\left(\begin{array}{ccc|cc} dim1 & dim2 & dim3 & dim4 & dim5 \\ 0.09 & 0.16 & -0.81 & 0.33 & -0.42 \\ 0.53 & 0.02 & -0.38 & -0.67 & 0.32 \\ 0.84 & -0.04 & 0.33 & 0.38 & -0.16 \\ 0.01 & 0.62 & -0.06 & 0.39 & 0.67 \\ 0.00 & 0.76 & 0.25 & -0.35 & -0.47 \end{array}\right) \left(\begin{array}{cc|ccc} 1.14 & 0 & 0 & 0 & 0 \\ 0 & 0.87 & 0 & 0 & 0 \\ \hline 0 & 0 & 0.64 & 0 & 0 \\ 0 & 0 & 0 & 0.35 & 0 \\ 0 & 0 & 0 & 0 & 0.16 \end{array}\right) \left(\begin{array}{ccc|cc} dim1 & dim2 & dim3 & dim4 & dim5 \\ 0.02 & 0.06 & -0.42 & 0.31 & -0.84 \\ 0.96 & -0.03 & 0.21 & 0.13 & -0.02 \\ 0.25 & 0.07 & -0.71 & -0.62 & 0.13 \\ 0.03 & 0.29 & -0.45 & 0.67 & 0.49 \\ 0.00 & 0.67 & 0.16 & -0.12 & -0.07 \\ 0.00 & 0.67 & 0.16 & -0.12 & -0.07 \end{array}\right)^T$$

C. Rank-2 Approximation Matrix

$$A_2 = U_2\Sigma_2 V_2^T \left(\begin{array}{c|cccccc} & swig & sip & drink & devour & eat & swallow \\ \hline beer & 0.0119 & 0.0959 & 0.0380 & 0.0466 & 0.0981 & 0.0981 \\ whiskey & 0.0174 & 0.5899 & 0.1598 & 0.0237 & 0.0153 & 0.0153 \\ coffee & 0.0233 & 0.9328 & 0.2470 & 0.0175 & -0.0206 & -0.0206 \\ bread & 0.0347 & -0.0079 & 0.0442 & 0.1639 & 0.3668 & 0.3668 \\ sugar & 0.0422 & -0.0206 & 0.0513 & 0.2007 & 0.4499 & 0.4499 \end{array}\right)$$

D. Two-dimensional plot of SVD Result

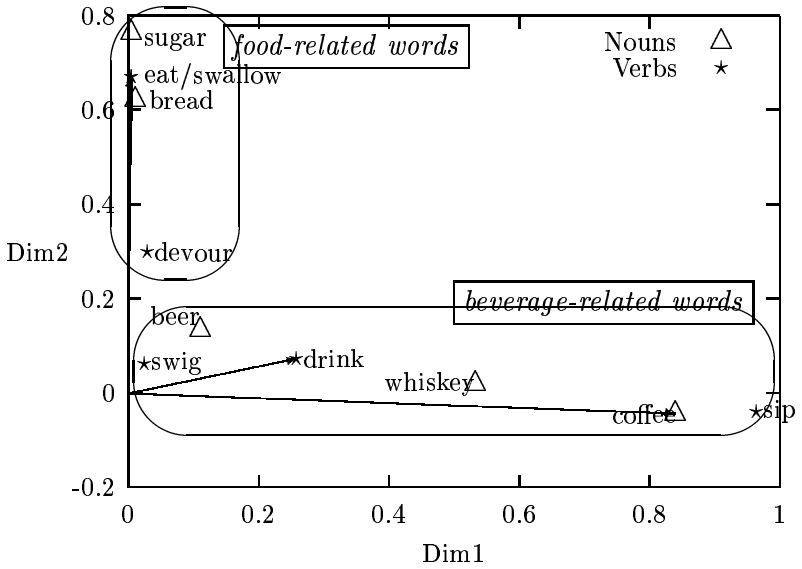

Figure 1: An Example of Singular Value Decomposition

Table 2: An Illustrative Example

	MLE	Katzs back-off	Similarity -based method ($k=3$)	Dimension Reduced Model		
				Distance -based ($dim=2$)	k-Rank matrix ($\delta=0.1$)	DR-SIM ($\theta=0$)
$P(swig \mid coffee)$	0	**0.09**	**0**	0.1627	0.0756	0.11
$P(sip \mid coffee)$	1	1	1	0.2425	0.5536	1
$P(drink \mid coffee)$	0	0.18	0.17	0.2359	0.1932	0.28
$P(devour \mid coffee)$	0	0.18	0.11	0.1271	0.0726	0.11
$P(eat \mid coffee)$	0	0.18	0.11	0.1159	0.0526	0
$P(swallow \mid coffee)$	0	0.18	0.11	0.1159	0.0526	0

2.3.3 Method 3: *Dimension-reduced similarity-based method*

The similarity-based method(Dagan et al., 1999) and dimension reduction technique can be merged into one model [3]. Reduced dimension can be better representation space than the original space for finding similarities between words.

This approach finds the most k nearest words to x in reduced space and use these word to estimate the probability $p(y|x)$:

$$P(y_i|x_i) = \begin{cases} P_{mle}(y_j|x_i) & c(x,y) > 0 \\ \frac{1}{|S|}\sum_{x_i' \in S} P_{mle}(y_j|x_i') & c(x,y) = 0 \end{cases} \quad (7)$$

where $S = \{x_i' | cos(\vec{x_i}, \vec{x_i}') > \theta \text{ on k dim.}\}$, the k is the reduced dimension size not a count of nearest nouns. The count of nearest nouns are determined by θ, which is threshold of cosine value.

[3] In the previous similarity-based work, (Dagan et al., 1999) used the complicated estimating equation:

$$\hat{P}(y_j|x_i) = \sum_{x' \in S(x,k)} \frac{W(x,x')}{Z(x)} \cdot P(y|x'),$$
$$W(x,x') = 10^{-\beta JS(x||x')}$$

where $W(x,x')$ is a similarity measure derived from the dissimilarity measure JS divergence. $S(n,k)$ is the set of k words with the smallest JS-divergence to x. Here, however, we use more simplifed equation:

$$\hat{P}(y_j|x_i) = \frac{1}{k} \sum_{x_i' \in S(x_i,k)} P(y_j|x_i')$$

3 An Illustrative Example

Here we use a concrete example to illustrate effectiveness of our model. The example is based on Table 1 and the task is estimating noun and verb co-occurrence probability $p(v|n)$. There are two groups of words, beverage-related words(*[beer, whicskey, coffee]/N, [swig, sip, drink]/V*) and food-related words(*[bread,sugar]/N, [devour,eat,swallow]/V*).

In Table 1, "coffee" is a sparsely distributed noun and we expect $P(drink|coffee) > P(eat|coffee)$. With MLE, it is not possible to rank two probabilities since they are all 0. Katz's back-off also fails to distinguish them since unigram probabilities $p(drink) = p(eat) = 0.18$. In the similarity-based scheme we compute JS-divergence to find similarity between nouns and $JS\big(p(v|beer), p(v|coffee)\big)$ [4] $= JS\big(p(v|bread), p(v|coffee)\big) = 0.6931$, which does not discriminate "beer" and "bread".

The distance-based model, however, solves all these problem. When we observe the third row in Figure 1-C, that is pre-normalized $p(v|coffee)$, there are no zero values unlike MLE. Futhermore, we can end up with two groups of verbs: beverage-related verbs have positive values $A_2(sip, coffee)$, $A_2(drink, coffee)$, $A_2(swig, coffee) > 0$ and food-related verbs have negative values $A_2(eat, coffee), A_2(swallow, coffee) < 0$.

[4] $JS(p,q) = \frac{1}{2}\big[D(p\|\frac{(p+q)}{2}) + D(q\|\frac{(p+q)}{2})\big]$, $D(p\|q) = \sum_t p(t) log \frac{p(t)}{q(t)}$

To concrete our example, $p(v|coffee)$ is constructed in Table 2 using probability estimation functions as described in the above section. MLE shows five zero values, causing data sparseness problems. Katz' back-off methods and similarity-based method cannot distinguish food-related verbs and drink-related verbs. In contrast, all dimension-reduced models resolve data sparseness problem and they cluster nouns and co-occurrence verbs reasonably. Thus, we can expect that dimension-reduced model will show promising result in a real experiment.

4 Experiment

We evaluated the dimension-reduced models on a pseudo word sense disambiguation task as in (Dagan et al., 1999). Each method is presented with a noun and two verbs, deciding which verb is more likely to have the noun as a direct object. Data preparation method and error counting scheme are almost similar to that of similarity-based methods (Dagan et al., 1999)(Lee and Pereira, 1999).

Performance is measured by the error rate, defined as

$$\text{error rate} = \frac{1}{T}(\sharp \text{ of incorrect choices})$$

where T is the size of test set. Test instances consist of noun-verb-verb triples $(n, v1, v2)$, where both $(n, v1)$ and $(n, v2)$ are both unseen in the training set. $(n, v1)$ is selected such that it appeared at least twice as often than $(n, v2)$ in the original verb-object pairs and $p(n, v1) > p(n, v2)$ is a correct answer. In addition, to consider Katzs back-off method as the baseline, $v2$ is choosed as $frq(v1) < frq(v2) \Leftrightarrow p(v1) < p(v2)$, and the error rate of back-off method is always 100% [5] . Running method is three-fold cross-validation and all results are averages over the three test sets.

[5]Katz's back-off estimator is defiend as the following eqaution. We set $\alpha(x_i) = 1$ here.

$$P_{bo}(y_j|x_i) = \begin{cases} P_d(y_j|x_i) & frq(x_i, y_j) > 0 \\ \alpha(x_i)P(y_j) & frq(x_i, y_i) = 0 \end{cases}$$

For similarity-based method, the parameter tuning is important to improve performance but we use the simplified unweighted average equation as in (Lee and Pereira, 1999) [6] . Since this equation is the same as our estimation method in Section 2.3.3, we can say that the comparison is fair. Number of similar nouns k is determined such that shows best result on test set.

4.1 Data Preparation

We prepared test sets as follows:

1. Extract transitive verb and head noun pairs from Penn Treebank II.
2. Select the pairs for the 1,000 most frequent nouns.
3. Partition the selected pairs 70% for training set and 30% for test set. (3 fold).
4. For each test set,
   (a) remove seen pairs.
   (b) for each $(n, v1)$, create $(n, v1, v2)$ such that $frq(n, v1) > 2 * frq(n, v2)$ and $frq(v1) < frq(v2)$.

Step 2 makes $p(v|n)$ matrix size fixed. Since it is difficult to find $(n, v1, v2)$ triples that satisfy Step 4-(b) criteria, average test set size is small. Hence, we used relative large portion, 30% of the pairs for building test set. Table 3 summarizes the experiment data.

Table 3: Training and Test Data.

1.Target Corpus	Penn Treebank II
3.Verb-object pairs	18843 pairs
3. $\|N\| \times \|V\|$ matrix size	1000×2008
4.Training set size	13040 pairs
5.Test set size	713 triples

4.2 Result

Table 4 shows the experimental error rate on the three test sets, using Katz's back-off as the baseline. Two dimension-reduced methods show much better performance than other methods.

[6]$\hat{p}(v|n) = \frac{1}{k} \sum_{n' \in S(n,k)} p(v|n')$

Table 4: Experimental Result (Error Rate)

	Katzs back-off (baseline)	Similarity -based method ($k = 20$)	Dimension Reduced Model		
			Distance -based ($dim = 90$)	k-Rank matrix ($\delta = 0.1$)	DR-SIM ($\theta = 0.5$)
Fold 1	1.0	0.623	0.362	0.386	0.586
Fold 2	1.0	0.636	0.374	0.423	0.594
Fold 3	1.0	0.645	0.366	0.402	0.593

The reason of such a good performance is that our model tries to find similarities between words toward two side (column and row space). The state-of-the-art similarity-based methods find similarities toward only one side. For example, their well-known similarity measures JS-divergence between given word x_i and another word x_i' is defined as $JS(p(y|x_i), p(y|x_i'))$. It means that each row $p(y|x_i)$ is compared to another row $p(y|x_i')$ in Table 1. Comparisons on column side are not performed. That's why the similarity-based fail to grasp true relationships on word co-occurrences.

On the other hand, SVD which is the mathematical background of our model gives us a reduced-rank basis for both column space and the row space simultaneously (Figure 1) (Michael W. Berry and Jessup, 1999). As we showed in the previous example in Section 3, the dimension-reduced model extract underlying or latent structures of word co-occurrences well. Therefore, our model shows successful result on estimating word co-occurrence probabilities of *sparse* data. futher However, the experiment is artificial and SVD is not directly related to the probabilty theory, futher theoretical invetigation is required.

4.3 Optimal subspace and Degree of sparseness

We also investigated the change of the performances as subspace size and degree of sparseness vary. Figure 2 shows performances of distance-based DR model as the dimension of subspace increase. When a dimension size is between 90 and 200, it shows the best result. Thus, we can conclude that the subspace of small dimension size ($90 \ll 1000$) is sufficent to capture latent word co-occurrence relationship.

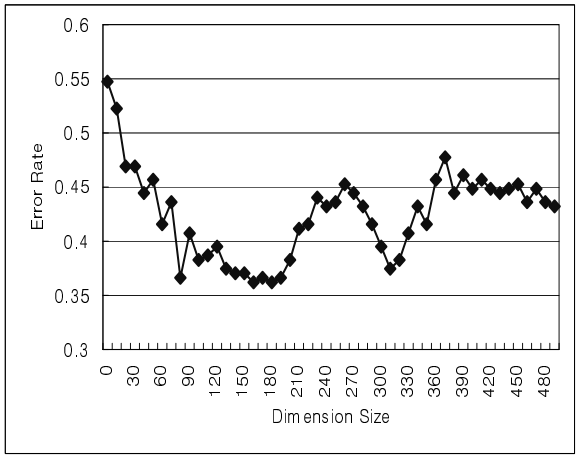

Figure 2: Performance vs. Dimension size

Figure 3 shows the effect of the degree of sparseness. The 1st ranked noun appears the most frequent times and 1000th ranked noun appears the least frequent times, in the training set. The average error rate does not change much as the sparseness increases. Therefore it is plausible to say that the dimension-reduced model does not show performance degration on very sparse data.

5 Conclusion

We proposed a novel approach called dimension-reduced estimation model for dealing with data sparseness problem. Three variant models are suggested and they are compared the performance against Katz's back-off method and similarity-based scheme. Dimension-reduced model can be alternative

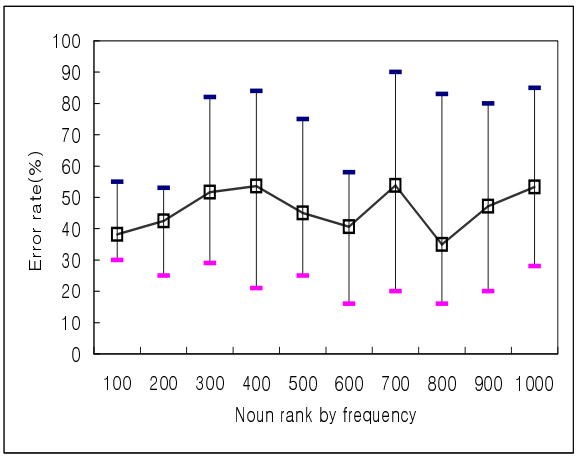

Figure 3: Performance vs. Degree of sparseness

probability smoothing scheme.

The ability of LSA that extracts and infers latent relations of words makes it possible to estimate probabilities of sparse data reasonably. LSA is a fully automatic mathematical technique. If we make a matrix from any given information once, we can use the reduced matrix for estimating probability. While the SVD analysis is somewhat costly in terms of time for large matrix, less expensive alternatives such as folding-in and SVD-updating have been suggested (Michael W. Berry and Jessup, 1999).

Further investigation is needed in both theoretical and experimental side. The suggested model does not have deep background over probablity theory. Hopefully, (Hofmann, 1999) suggested probabilistic LSI which is based on a statitical latent class model for factor analysis of count data. In addition, we applied our model to estimate bigram probabilities only. Corpus-based NLP is so mature and the methods must be tested with more realistic tasks. Since any conditional probability distributions can be represented by a matrix form, we can combines other information in a matrix, applying our model to more general tasks, such as word sense disambiguation and word clustering.

6 Acknowledgements

This work was supported by KOSEF through the "Multilingual Information Retrieval" project at the AITrc and was supported by Ministry of Culture and Tourism under the program of King Sejong Project through KORTERM. Many fundamental researches was supported by the R&D fund of Ministry of Science and Technology under a project of plan STEP2000.

References

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34:43–69.

Scott Deerwester, Susan Dumais, Goerge Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Susan T. Dumais, Michael L. Littman, and Thomas K. Landauer. 1997. Automatic cross-language retrieval using latent semantic indexing. In *In AAAI Spring Symposuim on Cross-Language Text and Speech Retrieval*, pages 18–24, Stanford University, March.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of SIGIR'99*, pages 50–57. ACM Press.

Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, mar.

Lillian Lee and Fernando Pereira. 1999. Distributional similarity models: Clustering vs. nearest neighbors. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Somerset, New Jersey. Association for Computational Linguistics.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Somerset, New Jersey. Association for Computational Linguistics.

Zlatko Drmac Michael W. Berry and Elizabeth R. Jessup. 1999. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362.