A Comparative Evaluation of Data-driven Models in Translation Selection of Machine Translation

∘Yu-Seop Kim ⋆ Jeong-Ho Chang ⋆Byoung-Tak Zhang
∘ Ewha Institute of Science and Technology, Ewha Woman's Univ.
Seoul 120-750 Korea, *yskim01@ewha.ac.kr**
⋆ Schools of Computer Science and Engineering, Seoul National Univ.
Seoul 151-742 Korea, {*jhchang, btzhang*}*@bi.snu.ac.kr*†

Abstract

We present a comparative evaluation of two data-driven models used in translation selection of English-Korean machine translation. Latent semantic analysis(LSA) and probabilistic latent semantic analysis (PLSA) are applied for the purpose of implementation of data-driven models in particular. These models are able to represent complex semantic structures of given contexts, like text passages. Grammatical relationships, stored in dictionaries, are utilized in translation selection essentially. We have used k-nearest neighbor (k-NN) learning to select an appropriate translation of the unseen instances in the dictionary. The distance of instances in k-NN is computed by estimating the similarity measured by LSA and PLSA. For experiments, we used TREC data(AP news in 1988) for constructing latent semantic spaces of two models and Wall Street Journal corpus for evaluating the translation accuracy in each model. PLSA selected relatively more accurate translations than LSA in the experiment, irrespective of the value of k and the types of grammatical relationship.

1 Introduction

Construction of language associated resources like thesaurus, annotated corpora, machine-readable dictionary and etc. requires high degree of cost, since they need much of human-effort, which is also dependent heavily upon human intuition. A data-driven model, however, does not demand any of human-knowledge, knowledge bases, semantic thesaurus, syntactic parser or the like. This model represents latent semantic structure of contexts like text passages. Latent semantic analysis (LSA) (Landauer et al., 1998) and probabilistic latent semantic analysis (PLSA) (Hofmann, 2001) fall under the model.

LSA is a theory and method for extracting and representing the contextual-usage meaning of words. This method has been mainly used for indexing and relevance estimation in information retrieval area (Deerwester et al., 1990). And LSA could be utilized to measure the coherence of texts (Foltz et al., 1998). By applying the basic concept, a vector representation and a cosine computation, to estimate relevance of a word and/or a text and coherence of texts, we could also estimate the semantic similarity between words. It is claimed that LSA represents words of similar meaning in similar ways (Landauer et al., 1998).

Probabilistic LSA (PLSA) is based on probabilistic mixture decomposition while LSA is on a linear algebra and singular value decomposition (SVD) (Hofmann, 1999b). In contrast to LSA, PLSA's probabilistic variant has a sound statistical foundation and defines a proper generative model of the data. Both two techniques have a same idea which is to map high-dimensional vectors representing text documents, to a lower dimensional representation, called a latent semantic space (Hofmann, 1999a).

Dagan (Dagan et al., 1999) performed a comparative analysis of several similarity measures, which based mainly on conditional probability distribution. And the only elements in the distribution are words, which appeared in texts. However, LSA and PLSA expressed the latent semantic structures, called a topic of the context.

In this paper, we comparatively evaluated these two techniques performed in translation

* He is supported by Brain Korea 21 project performed by Ewha Institute of Science and Technology.
† They are supported by Brain Tech. project controlled by Korean Ministry of Science and Technology.

selection of English-Korean machine translation. First, we built a dictionary storing tuples representing the grammatical relationship of two words, like *subject-verb*, *object-verb*, and *modifier-modified*. Second, with an input tuple, in which an input word would be translated and the other would be used as an argument word, translation is performed by searching the dictionary with the argument word. Third, if the argument word is not listed in the dictionary, we used k-nearest neighbor learning method to determine which class of translation is appropriate for the translation of an input word. The distance used in discovering the nearest neighbors was computed by estimating the similarity measured on above latent semantic spaces.

In the experiment, we used 1988 AP news corpus from TREC-7 data (Voorhees and Harman, 1998) for building latent semantic spaces and Wall Street Journal (WSJ) corpus for constructing a dictionary and test sets. We obtained 11-20% accuracy improvement, comparing to a simple dictionary search method. And PLSA has shown that its ability to select an appropriate translation is superior to LSA as an extent of up to 3%, without regard to the value of k and grammatical relationship.

In section 2, we discuss two of data-driven models, LSA and PLSA. Section 3 describes ways of translation with a grammatical relation dictionary and k-nearest neighbor learning method. Experiment is explained in Section 4 and concluding remarks are presented in Section 5.

2 Data-Driven Model

For the data-driven model which does not require additional human-knowledge in acquiring information, Latent Semantic Analysis (LSA) and Probabilistic LSA (PLSA) are applied to estimate semantic similarity among words. Next two subsections will explain how LSA and PLSA are to be adopted to measuring semantic similarity.

2.1 Latent Semantic Analysis

The basic idea of LSA is that the aggregate of all the word contexts in which a given word does and does not appear provides a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other (Landauer et al., 1998)(Gotoh and Renals, 1997). LSA also extracts and infers relations of expected contextual usage of words in passages of discourse. It uses no human-made dictionaries, knowledge bases, semantic thesaurus, syntactic parser or the like. Only raw text parsed into unique character strings is needed for its input data.

The first step is to represent the text as a matrix in which each row stands for a unique word and each column stands for a text passage or other context. Each cell contains the occurrence frequency of a word in the text passage.

Next, LSA applies singular value decomposition (SVD) to the matrix. SVD is a form of factor analysis and is defined as

$$A = U\Sigma V^T \tag{1}$$

,where Σ is a diagonal matrix composed of nonzero eigen values of AA^T or A^TA, and U and V are the orthogonal eigenvectors associated with the r nonzero eigenvalues of AA^T and A^TA, respectively. One component matrix (U) describes the original row entities as vectors of derived orthogonal factor value, another (V) describes the original column entities in the same way, and the third (Σ) is a diagonal matrix containing scaling values when the three components are matrix-multiplied, the original matrix is reconstructed.

The singular vectors corresponding to the $k(k \leq r)$ largest singular values are then used to define k-dimensional document space. Using these vectors, $m \times k$ and $n \times k$ matrices U_k and V_k may be redefined along with $k \times k$ singular value matrix $\sum_k$. It is known that $A_k = U_k \Sigma_k V_k^T$ is the closest matrix of rank k to the original matrix.

LSA can represent words of similar meaning in similar ways. This can be claimed by the fact that one compares words with similar vectors as derived from large text corpora. The term-to-term similarity is based on the inner products between two row vectors of A, $AA^T = U\Sigma^2U^T$. One might think of the rows of $U\Sigma$ as defining coordinates for terms in the latent space. To calculate the similarity of coordinates, $\mathbf{V_1}$ and $\mathbf{V_2}$, cosine computation is used:

$$\cos\phi = \frac{\mathbf{V_1} \cdot \mathbf{V_2}}{\| \mathbf{V_1} \| \cdot \| \mathbf{V_2} \|} \tag{2}$$

2.2 Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis (PLSA) is a statistical technique for the analysis of two-mode and co-occurrence data, and has produced some meaningful results in such applications as language modelling (Gildea and Hofmann, 1999) and document indexing in information retrieval (Hofmann, 1999b). PLSA is based on *aspect model* where each observation of the co-occurrence data is associated with a latent class variable $z \in Z = \{z_1, z_2, \ldots, z_K\}$ (Hofmann, 1999a). For text documents, the observation is an occurrence of a word $w \in W$ in a document $d \in D$, and each possible state z of the latent class represents one semantic topic.

A word-document co-occurrence event, (d, w), is modelled in a probabilistic way where it is parameterized as in

$$\begin{aligned} P(d,w) &= \sum_z P(z)P(d,w|z) \\ &= \sum_z P(z)P(w|z)P(d|z). \end{aligned} \quad (3)$$

Here, w and d are assumed to be conditionally independent given a specific z. $P(w|z)$ and $P(d|z)$ are topic-specific word distribution and document distribution, respectively. The three-way decomposition for the co-occurrence data is similar to that of SVD in LSA. But the objective function of PLSA, unlike that of LSA, is the likelihood function of multinomial sampling. And the parameters $P(z), P(w|z)$, and $P(d|z)$ are estimated by maximization of the log-likelihood function

$$L = \sum_{d \in D} \sum_{w \in W} n(d,w) \log P(d,w), \quad (4)$$

and this maximization is performed using the EM algorithm as for most latent variable models. Details on the parameter estimation are referred to (Hofmann, 1999a). To compute the similarity of w_1 and w_2, $P(z_k|w_1)P(z_k|w_2)$ should be approximately computed with being derived from

$$P(z_k|w) = \frac{P(z_k)P(w|z_k)}{\sum_{z_k \in Z} P(z_k)P(w|z_k)} \quad (5)$$

And we can evaluate similarities with the low-dimensional representation in the semantic topic space $P(z_k|w_1)$ and $P(z_k|w_2)$.

3 Translation with Grammatical Relationship

3.1 Grammatical Relationship

We used grammatical relations stored in the form of a dictionary for translation of words. The structure of the dictionary is as follows (Kim and Kim, 1998):

$$T(S_i) = \begin{cases} T_1 & \text{if } Cooc(S_i, S_1) \\ T_2 & \text{if } Cooc(S_i, S_2) \\ & \ldots \\ T_n & \text{otherwise,} \end{cases} \quad (6)$$

where $Cooc(S_i, S_j)$ denotes grammatical co-occurrence of source words S_i and S_j, which one means an input word to be translated and the other means an argument word to be used in translation, and T_j is the translation result of the source word. $T(\cdot)$ denotes the translation process.

Table 1 shows a grammatical relationship dictionary for an English verb S_i = *'build'* and its object nouns as an input word and an argument word, respectively. The dictionary shows that the word *'build'* is translated into five different translated words in Korean, depending on the context. For example, *'build'* is translated into 'geon-seol-ha-da' ('construct') when its object noun is a noun 'plant' (='factory'), into 'che-chak-ha-da' ('produce') when co-occurring with the object noun 'car', and into 'seol-lip-ha-da' ('establish') in the context of object noun 'company' (Table 2).

One of the fundamental difficulties in co-occurrence-based approaches to word sense disambiguation (translation selection in this case) is the problem of data sparseness or unseen words. For example, for an unregistered object noun like 'vehicle' in the dictionary, the correct translation of the verb cannot be selected using the dictionary described above. In the next subsection, we will present k-nearest neighbor method that resolves this problem.

3.2 k-Nearest Neighbor Learning for Translation Selection

The similarity between two words on latent semantic spaces is required when performing k-NN search to select the translation of a word. The nearest instance of a given word is decided by selecting a word with the highest similarity to the given word.

Table 1: Examples of co-occurrence word lists for a verb *'build'* in the dictionary

Meaning of *'build'* in Korean (T_j)	Collocated Object Noun (S_j)			
'geon-seol-ha-da' (= 'construct')	plant	facility	network	...
'geon-chook-ha-da' (= 'design')	house	center	housing	...
'che-chak-ha-da' (= 'produce')	car	ship	model	...
'seol-lip-ha-da' (= 'establish')	company	market	empire	...
'koo-chook-ha-da' (= 'develop')	system	stake	relationship	...

Table 2: Examples of translation of *'build'*

source words		translated words (in Korean)	sense of the verb
'**build** a *plant*'	⇒	'*gong-jang*-eul **geon-seol-ha-da**'	**'construct'**
'**build** a *car*'	⇒	'*ja-dong-cha*-reul **che-chak-ha-da**'	**'produce'**
'**build** a *company*'	⇒	'*hoi-sa*-reul **seol-lip-ha-da**'	**'establish'**

The k-nearest neighbor learning algorithm (Cover and Hart, 1967)(Aha et al., 1991) assumes all instances correspond to points in the n-dimensional space R^n. We mapped the n-dimensional space into the n-dimensional vector of a word for an instance. The nearest neighbors of an instance are defined in terms of the standard Euclidean distance.

Then the distance between two instances x_i and x_j, $D(x_i, x_j)$, is defined to be

$$D(x_i, x_j) = \sqrt{(a(x_i) - a(x_j))^2} \qquad (7)$$

and $a(x_i)$ denotes the value of instance x_i, similarly to cosine computation between two vectors. Let us consider learning discrete-valued target functions of the form $f : R^n \rightarrow V$, where V is the finite set $\{v_1, \ldots, v_s\}$. The k-nearest neighbor algorithm for approximating a discrete-valued target function is given in Table 3.

The value $\hat{f}(x_q)$ returned by this algorithm as its estimate of $f(x_q)$ is just the most common value of f among the k training examples nearest to x_q.

Table 3: The k-nearest neighbor learning algorithm.

- Training
  - For each training example $\langle x, f(x)\rangle$, add the example to the list *training_examples*.
- Classification
  - Given a query instance x_q to be classified,
    - Let $x_1, \ldots, x_k$ denote the k instances from *training_examples* that are nearest to x_q.
    - Return

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i)) \ ,$$

where $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ otherwise.

4 Experiment and Evaluation

4.1 Data for Latent Space and Dictionary

In the experiment, we used two kinds of corpus data, one for constructing LSA and PLSA spaces and the other for building a dictionary containing grammatical relations and a test set. 79,919 texts in 1988 AP news corpus from TREC-7 data was indexed with a stemming tool and 19,286 words with the frequency of above 20 are extracted. We built 200 dimensions in SVD of LSA and 128 latent dimensions of PLSA. The difference of the numbers was caused from the degree of computational complexity in learning phase. Actually, PLSA of 128 latent factors required 50-fold time as much as LSA hiring 200 eigen-vector space during building latent spaces. This was caused by 50 iterations which made the log likelihood maximized. We utilized a sin-

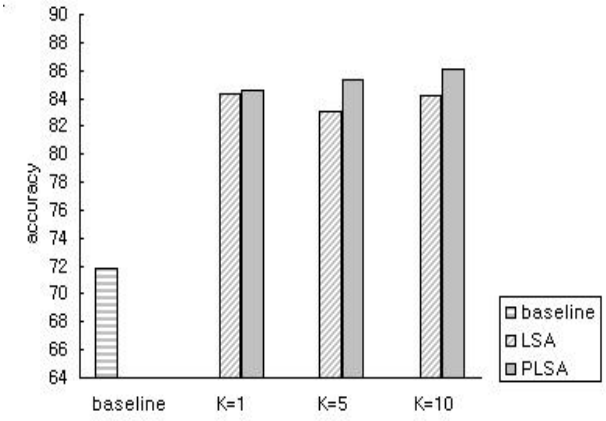

Figure 1: The accuracy ration of *verb-object*

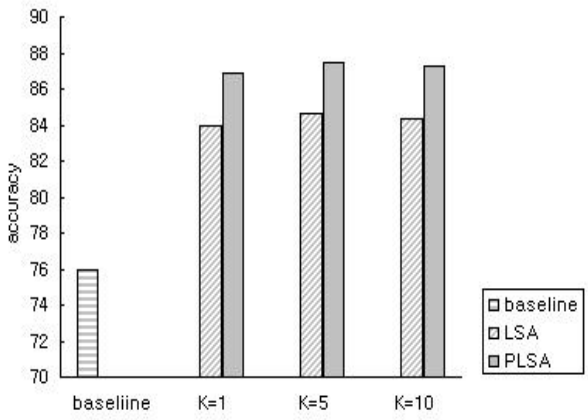

Figure 2: The accuracy ration of *subject-verb*

gle vector lanczos algorithm derived from SVDPACK when constructing LSA space. (Berry et al., 1993). We generated both of LSA and PLSA spaces, with each word having a vector of 200 and 128 dimensions, respectively. The similarity of any two words could be estimated by performing cosine computation between two vectors representing coordinates of the words in the spaces.

Table 4 shows 5 most similar words of randomly selected words from 3,443 examples. We extracted 3,443 example sentences containing grammatical relations, like *verb-object*, *subject-verb* and *adjective-noun*, from Wall Street Journal corpus of 220,047 sentences and other newspapers corpus of 41,750 sentences, totally 261,797 sentences. We evaluated the accuracy performance of each grammatical relation. 2,437, 188, and 818 examples were utilized for *verb-object*, *subject-verb*, and *adjective-noun*, respectively. The selection accuracy was measured using 5-fold cross validation for each grammatical relation. Sample sentences of each grammatical relation were divided into five disjoint samples and each sample became a test sample once in the experiment and the remaining four samples were combined to make up a collocation dictionary.

4.2 Experimental Result

Table 5 and figure 1-3 show the results of translation selection with respect to the applied model and to the value of k. As shown in Table 5, similarity based on data-driven model could improve the selection accuracy up to 20% as

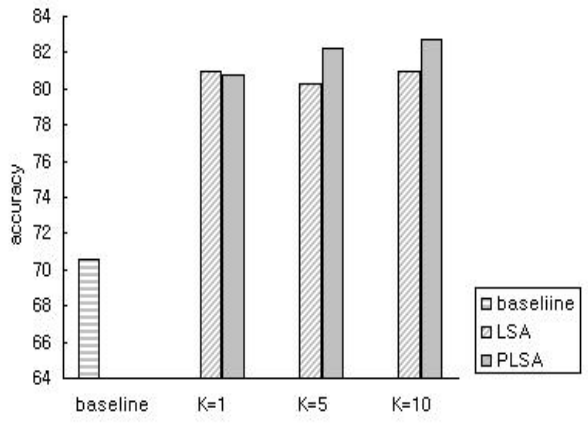

Figure 3: The accuracy ration of *adjective-noun*

contrasted with the direct matching method. We could obtain the result that PLSA could improve the accuracy more than LSA in almost all cases. The amount of improvement is varied from -0.12% to 2.96%.

As figure 1-3 show, the value of k had affection to the translation accuracy in PLSA, however, not in LSA. From this, we could not declare whether the value of k and translation accuracy have relationship of each other or not in the data-driven models described in this paper. However, we could also find that the degree of accuracy was raised in accordance with the value of k in PLSA. From this, we consequently inferred that the latent semantic space generated by PLSA had more sound distribution with reflection of well-structured semantic structure than LSA. Only one of three grammatical re-

Table 4: Lists of 5 most semantically similar words for randomly selected words generated from LSA, and PLSA. The words are stems of original words. The first row of each selected word stands for the most similar words in LSA semantic space and the second row stands for those in the PLSA space.

selected words	most similar words				
plant	westinghous	isocyan	shutdown	zinc	manur
	radioact	hanford	irradi	tritium	biodegrad
car	buick	oldsmobil	chevrolet	sedan	corolla
	highwai	volkswagen	sedan	vehicular	vehicle
home	parapleg	broccoli	coconut	liverpool	jamal
	memori	baxter	hanlei	corwin	headston
business	entrepreneur	corpor	custom	ventur	firm
	digit	compat	softwar	blackston	zayr
ship	vessel	sail	seamen	sank	sailor
	destroy	frogmen	maritim	skipper	vessel

Table 5: Translation accuracy in various case. The first column stands for each grammatical relation and the second column stands for the used models, LSA or PLSA. And other three columns stand for the accuracy ratio (r_m) with respect to the value of k. The numbers in parenthesis of the first column show the translation accuracy ratio of simple dictionary search method (r_s). And numbers in the other parenthesis were obtained by $r_m \div r_s$.

grammatical relations	used model	$k = 1$	$k = 5$	$k = 10$
verb-object	LSA	84.41(1.17)	83.01(1.16)	84.24(1.17)
(71.85)	PLSA	84.53(1.18)	85.35(1.19)	86.05(1.20)
subject-verb	LSA	83.99(1.11)	84.62(1.11)	84.31(1.11)
(75.93)	PLSA	86.85(1.14)	87.49(1.15)	87.27(1.15)
adjective-noun	LSA	80.93(1.15)	80.32(1.14)	80.93(1.15)
(70.54)	PLSA	80.81(1.15)	82.27(1.17)	82.76(1.17)

lations, *subj-verb*, showed an exceptional case, which seemed to be caused by the small size of examples, 188.

Selection errors taking place in LSA and PLSA models were caused mainly by the following reasons. First of all, the size of vocabulary should be limited by computation complexity. In this experiment, we acquired below 20,000 words for the vocabulary, which could not cover a section of corpus data. Second, the stemming algorithm was not robust for an indexing. For example, *'house'* and *'housing'* are regarded as a same word as *'hous'*. This fact brought about hardness in reflecting the semantic structure more precisely. And finally, the meaning of *similar word* is somewhat varied in the machine translation field and the information retrieval field. The selectional restriction tends to depend a little more upon semantic type like *human-being*, *place* and etc., than on the context in a document.

5 Conclusion

This paper describes a comparative evaluation of the accuracy performance in translation selection based on data-driven models. LSA and PLSA were utilized for implementation of the models, which are mainly used in estimating similarity between words. And a manually-built grammatical relation dictionary was used for the purpose of appropriate translation selection of a word. To break down the data sparseness problem occurring when the dictionary is used, we utilized similarity measurements schemed out from the models. When an argument word is not included in the dictionary, the most k similar words to the word are discovered in the dictionary, and then the meaning of

the grammatically-related class for the majority of the k words is selected as the translation of an input word.

We evaluated the accuracy ratio of LSA and PLSA comparatively and classified the experiments with criteria of the values of k and the grammatical relations. We acquired up to 20% accuracy improvement, compared to direct matching to a collocation dictionary. PLSA showed the ability to select translation better than LSA, up to 3%. The value of k is strongly related with PLSA in translation accuracy, not too with LSA. That means the latent semantic space of PLSA has more sound distribution of latent semantics than that of LSA. Even though longer learning time than LSA, PLSA is beneficial in translation accuracy and distributional soundness. A distributional soundness is expected to have better performance as the size of examples is growing.

However, we should resolve several problems raised during the experiment. First, a robust stemming tool should be exploited for more accurate morphology analysis. Second, the optimal value of k should be obtained, according to the size of examples. Finally, we should discover more specific contextual information suited to this type of problem. While simple text could be used properly in IR, MT should require another type of information.

The data-driven models could be applied to other sub-fields related with semantics in machine translation. For example, to-infinitive phrase and preposition phrase attachment disambiguation problem can also apply these models. And syntactic parser could apply the models for improvement of accurate analysis by using semantic information generated by the models.

References

D. Aha, D. Kibler, and M. Albert. 1991. Instance-based learning algorithms. *Machine Learning*, 6:37–66.

M. Berry, T. Do, G. O'Brien, V. Krishna, and S. Varadhan. 1993. Svdpackc: Version 1.0 user's guide. Technical Report CS–93–194, University of Tennessee, Knoxville, TN.

T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

I. Dagan, L. Lee, and F. Fereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34:43–69.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

P. Foltz, W. Kintsch, and T. Landauer. 1998. The mesurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25:285–307.

D. Gildea and T. Hofmann. 1999. Topic based language models using em. In *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech99)*.

D. Gotoh and S. Renals. 1997. Document space models using latent semantic analysis. In *Proceedings of Eurospeech-97*, pages 1443–1446.

T. Hofmann. 1999a. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI'99)*.

T. Hofmann. 1999b. Probabilistic latent semantic indexing. In *Proceedings of the 22th Annual International ACM SIGIR conference on Research and Developement in Information Retrieval (SIGIR99)*, pages 50–57.

T. Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196.

Y. Kim and Y. Kim. 1998. Semantic implementation based on extended idiom for english to korean machine translation. *The Asia-Pacific Association for Machine Translation Journal*, 21:23–39.

T. K. Landauer, P. W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

E. Voorhees and D. Harman. 1998. Overview of the seventh text retrieval conference (trec-7). In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 1–24.