Pacific Association for Computational Linguistics

INFORMATION RETRIEVAL BASED ON DOMAIN-SPECIFIC WORD ASSOCIATIONS

YASUHIRO TAKAYAMA
Information Technology R & D Center, Mitsubishi Electric Corporation
5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501, Japan
RAYMOND FLOURNOY, STEFAN KAUFMANN, STANLEY PETERS
Center for the Study of Language and Information, Stanford University
210 Panama Street, Stanford, CA 94305-4115, USA

The text corpus for information retrieval (IR) consists of a collection of words, so representation of word senses is an essential basis for IR. We have developed representation of word senses as word vectors derived from text corpora. We have to concern the dimensionality in order to use word vectors for applications, so we use singular value decomposition (SVD) as a dimensionality reduction tool. Furthermore we extend word vectors to context vectors and construct IR system called InfoMap. This paper describes how to create word vectors, the relation between SVD and Principal Component Analysis, the configuration of InfoMap system, and a preliminary experimental result using a domain-specific English text corpus OHSUMED.

Key words: natural language base software, word sense representation, information retrieval

INTRODUCTION

The text corpus targeted for information retrieval consists of a collection of words. So representation of word sense is an essential basis for IR. There are various studies that represent word senses as multidimensional word vectors. One uses dictionaries as base information of word vectors, and another uses co-occurrence relation in corpus [Niwa 93]. In both case, there's a problem about size of dimensionality of word vectors when we apply word vectors to application software such as IR systems.

[Kozima 97] described a method for calculation of word sense distance based on dictionary. [Kozima 97] calculates vector space based on weights among definition words of LDOCE (Longman Dictionary of Contemporary English) by spread activation and reduce the vector space by principal component analysis to construct word sense space. However, we cannot assume the existence of a well-defined dictionary like LDOCE when we process domain specific corpora.

Thus, we have investigated a method to construct word vectors based on co-occurrence relation based on corpora. To reduce the dimensionality of co-occurrence vectors, we construct word sense space called **Word Space** by using singular value decomposition (SVD). This paper describes how to construct Word Space and explains SVD calculation is equivalent to Principal Component Analysis in multivariate analysis. Then, we expand the word sense representation Word Space to context vectors and an IR system called **InfoMap** (Information Mapping). Furthermore, we report some preliminary experimental result using domain specific corpus OHSUMED [Hersh 94] in InfoMap system.

1. ASSOCIATIVE IR AND WORD SPACE

1.1 IR based on Word Associations

Currently, full text document retrieval from large text databases is based on keyword search. A query is posed as a list of words, and any entries in the database, which contain any or all of those specific words, are returned. However, if we treat those query words not as literal strings of letters, but as representing *concepts*, then we can retrieve relevant documents even if they do not contain the specific words used in the query. The goal of our InfoMap project is intelligent, concept-based IR.

Our basic approach, developed by Hinrich Schütze [Schütze 95], begins by recording the frequency of co-occurrence between words in the text; that is, the number of times two words appear "near" each other, e.g., in the same document. The distribution of co-occurrences between a word and some set of *content-bearing words* then serves as a profile of the word's usage, and thus of its meaning as well.

By comparing the profiles of different words, we can construct a similarity measure of how related those words are. Generalizing this word similarity derived from lexical co-occurrence, by comparing the query words' profiles to profiles generated for each document, we can return documents which we judge to be conceptually related to the query words, even if the words themselves do not appear in the text.

The retrieval methods based on vector space models are called "conceptual retrieval". We call our method *associative information retrieval* that is based on word associations.

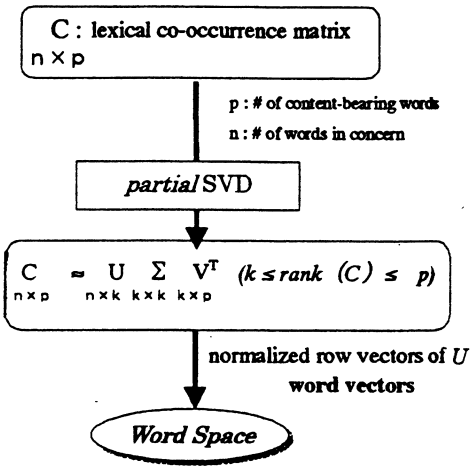

FIGURE 1. SVD for Word Space

1.2 Word Space calculated by SVD

The lexical co-occurrences between a word and content-bearing words are recorded in the *co-occurrence matrix* which creates a high-dimensional space. This abstract space forms a concept space in which similar words (or more specifically, words with similar *distributional* behavior) have similar vectors (See Table 1).

The co-occurrence matrix suffers from two problems: too many word features and data sparseness. To solve these problems, we apply SVD (Singular Value Decomposition) [Strang 93] to the co-occurrence matrix as a tool for dimensionality reduction and generalization. SVD factors every m by n matrix A into

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n}. \quad (1)$$

Where the left matrix U and the right matrix V are orthogonal matrices and the singular matrix Σ is diagonal. The superscript T denotes transposition.

TABLE 1. An example of co-occurrence matrix

content-bearing words / words	...	market	...	last	...	
		1000 words				
Sunday		97	...	215	...	20k words
...	...	...	...	...	...	
weekend		201	...	408	...	
...	...	...	...	...	...	

The equation (1) shows the *full* SVD in linear algebra. We use the left orthogonal matrix U as the reduced matrix, the output from the *partial* SVD (Figure 1). The rows of the reduced matrix — *word vectors* — approximate *associations* among the word senses. This reduced space from the previous concept space is called Word Space. It potentially reflects associative behavior of words captured through *second-order co-occurrence* information. Another use of SVD in information retrieval is word by document matrix reduction for LSI (Latent Semantic Indexing) [Deerwester 90]. The difference between Word Space and LSI is discussed in [Schütze 97]. By clustering the word vectors based on their proximity, the Word Space can be used for the word sense disambiguation and thesaurus construction [Schütze 95, 98].

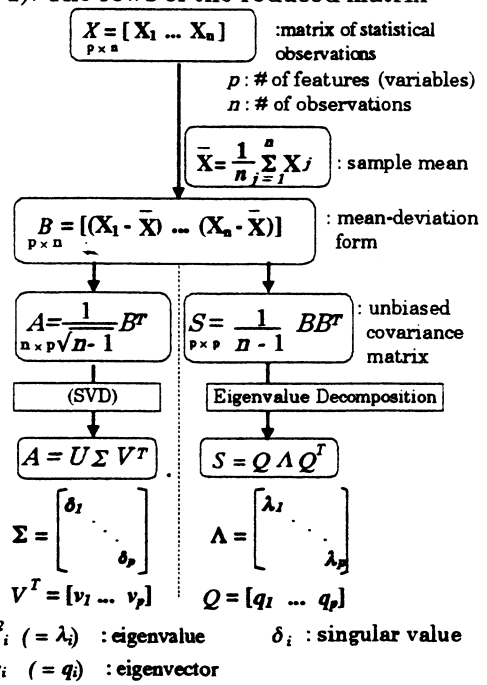

FIGURE 2. Relation between SVD and PCA

1.3 SVD and Principal Component Analysis

SVD is not a direct statistical technique but rather a matrix factorization in linear algebra. When a matrix to be processed consists of some statistical observations, SVD becomes a powerful tool for statistical analysis. SVD has a close relationship with the *principal component analysis* (PCA), a feature reduction technique in multivariate analysis [Lay 97][Schütze 95]. Multivariate analysis concerns associations among multiple variables (features) with the goal of discovering relationships among the multvariate profiles of the data.

Suppose that matrix X is a $p \times n$ matrix of observations (or a data matrix). If matrix B is a

matrix in mean-deviation form of the data matrix X, and if $A = (1/\sqrt{(n-1)})\, B^T$, then $A^T A$ becomes the *unbiased* covariance matrix S. We can calculate the eigenvalues and the eigenvectors by the eigenvalue decomposition from the $p \times p$ covariance matrix S. Eigenvalue decomposition can be applied to the square matrices only, but SVD can be applied to any rectangular matrices. Thus the calculation of the SVD is more convenient than eigenvalue decomposition.

When we apply SVD to the matrix A, the square of the singular values of A are the p eigenvalues of the covariance matrix S, and the right singular vectors $v_1 \cdots v_p$ of A are the coefficients of the principal components of the data in the matrix X. Then $v_i^T X$ is the i-th principal component. SVD can be used as a tool for performing the PCA (See Figure 2). In **Word Space**, we directly applies SVD to the original data matrix (i.e. lexical co-occurrence matrix C in our case) instead of the matrix A, the mean-deviation form with a coefficient $1/\sqrt{(n-1)}$ (See Figure 1).

2. SYSTEM ORGANIZATION OF INFOMAP SYSTEM

The retrieval model of the **InfoMap** search engine is based on a *vector space model* [Salton 75], that is, the documents and the queries are represented as vectors in the high-dimensional, just as the words are. The **InfoMap** search engine consists of the document registration phase that creates the **Word Space** (concept base) and the document retrieval phase, similar to other IR systems. This section illustrates the functions of these phases.

2.1 Construction of Word Space based on lexical co-occurrence

The document registration phase of **InfoMap** is the **Word Space** (concept base) construction functions based on lexical co-occurrence in the text corpus (Figure 3).

2.1.1 Tokenization and word frequency calculation

The first stage of processing produces a tokenized corpus. The corpus can be tokenized by passing it through a *tokenizer*. The *stemming* [Porter 80] in the tokenizer is optional. The second stage of processing produces a *word count dictionary*. The count dictionary is a word list of tokens and their frequencies in the corpus, ordered by frequency of appearance of the tokens.

2.1.2 Calculation of co-occurrence frequencies

For each of the 20,000[1] most frequently occurring words in the corpus, a vector of 1,000 co-occurrence counts is created, and these vectors serve as profiles of each word's distribution. The 1,000 entries in the vector represent a set of 1,000 words, which have been determined to be *content-bearing* in the following sense.

The content-bearing words are chosen by considering either the word's total frequency of appearance in the corpus, the word's part-of-speech information, or a calculation of the relative concentration of the word within the documents in the corpus.

This calculation -- called the "dispersion" of a word -- exploits the idea that words, which are not distributed evenly throughout the documents in a corpus, are more likely to be content-bearing. We choose the 51 to 1,050 most frequently occurring words in the corpus as a basic set of the content-bearing words.

Each time one of the 20,000 count words appears within a *window* -- a specific range of one of the content-bearing words, the appropriate count in its vector is incremented. A word falls within range if it is within a certain distance from the content-bearing word, or if it is within the same sentence, paragraph, or document as the content-bearing word.

After all documents in the corpus have been processed, the square root of each count is taken to smooth out the effects of extreme numbers. So the actual (i,j)-th element of the co-occurrence matrix is represented by a real value:

$$c_{ij} = \phi(cooc_{ij}) \qquad (2)$$

where $cooc_{ij}$ is the co-occurrence count of word i within a window from a content-bearing word j

[1] The numbers of the dimensions in this paper are example ones used in our experiment. Setting the parameters in the system configuration can change them.

throughout the corpus, and ϕ is the transformation of the count data. We use the square root as the basic transformation. Our standard setting of the window size is 51(25 words to the left and to the right of the current words).

2.1.3 Analysis of the second-order co-occurrence

The 20,000 vectors (the rows of the co-occurrence matrix) represent points in a 1,000-dimensional space. To make computations using the concept space more tractable, it is necessary to lower the dimensionality of the space. The tool we use for reducing dimensionality of a matrix of co-occurrence count is SVD [Lay 97].

This calculation is done by feeding the matrix through the SVDPack [Berry 92], a process which iteratively extracts the most important dimensional features to approximate the high-dimensional space with one of a much lower dimensionality. The left orthogonal matrix U, the output of the partial SVD in Figure 1 is now reduced to $p = 100$ dimensions. The row vectors of the reduced matrix, i.e. the left vectors serve as the word vectors u_i ($i = 1, ..., 20{,}000$) in Word Space derived from the lexical co-occurrence.

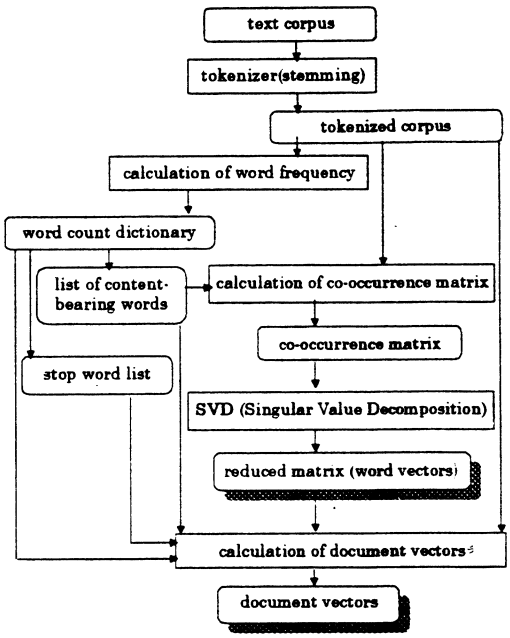

FIGURE 3. Word Space Construction

2.1.4 Creation of document vectors on Word Space

Each document is processed into a *document vector* of length 100. This is done by reading in the individual word vectors previously calculated for the 20,000 most frequently occurring words in the corpus, and summing the normalized vectors corresponding to each of the words in the document:

$$d_j = \sum_i w_{ij} u_i \qquad (3)$$

where d_j is the document vector for document j, w_{ij} is the weight for word i in document j, and u_i is the word vector for word i occurred in document j. The default weight w_{ij} is 1. The tf · idf (term frequency · inverse document frequency) weight is used in [Schütze 97].

Optionally, one may choose the *stop words* to disregard the vectors of certain words that are expected to be so general or so common that they will not contribute informatively to the vector. We use the 1 to 50 most frequently occurring words in the corpus as a basic set of the stop words.

After document vectors are calculated for each of the documents in the corpus, they are written to disk with the byte location of the document. The 100-dimensional space which these vectors occupy embodies the document concept base derived from the corpus, and each of these vectors represents a specific location within this space corresponding to the meaning or subject matter of the document. Furthermore, the formalism predicts that vectors which lie close to each other in the concept space correspond to documents which are somehow related in subject matter.

For simplicity, our explanation has included individual words as the dimensions of the co-occurrence matrix. Optionally, we also choose the statistically significant phrases based on χ^2-test that is applied to a contingency table of the neighboring word counts [Schütze98]. To find the pairs that most frequently "stick" together, we count all neighbor words, and sort them by frequency, then calculate their χ^2-value. A certain number (e.g. 5,000) of the top χ^2-valued words are considered *sticky pairs*. We also allow these sticky pairs to be elements for the row dimension of the co-occurrence matrix.

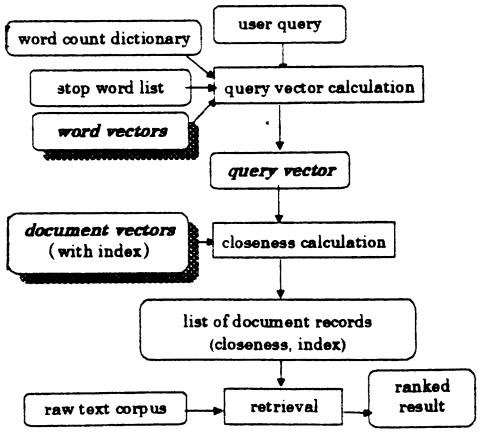

FIGURE 4. Document Retrieval on Word Space

2.2 Document retrieval on Word Space

The main stages of the document retrieval phase of **InfoMap** are the query vector calculation, the closeness (similarity) calculation and the actual retrieval (Figure 4).

2.2.1 Query vector calculation

To retrieve the documents from the corpus using the associations in **Word Space**, a query in the form of a list of words (either entered interactively or stored in a file) is translated into the corresponding set of word vectors, and these are summed to form a query vector:

$$q = \sum_i w_i u_i \qquad (4)$$

where q is the query vector, w_i is the weight for word i in the query (default weight is 1), and u_i is the word vector for word i occurred in the query.

2.2.2 Closeness calculation

The query vector is then compared with each of the document vectors and the documents whose vectors lie closest to the query vector are returned. Query vectors and document vectors are represented as normalized word vector sums (centroids). These vectors are called *context vectors* in general.

The *closeness* of two context vectors (the query vector q and a document vector d_j) is determined by calculating the cosine of the angle between the vectors[2]:

$$closeness(q, d_j) = (q \cdot d_j) / (|q| \cdot |d_j|) \qquad (5)$$

This routine requires the tokenized corpus and the document vectors (d_j) as its input and returns a linked-list of document records, ordered by closeness (similarity) with the query vector (q). Each document record contains the cosine score for the document and the byte location of the document in the corpus.

The retrieval routine simply seeks the location in the document records and displays the documents as requested by the user.

3. EXPERIMENT FOR DOMAIN-SPECIFIC CORPUS

As we described in previous sections, word associations in **Word Space** are derived from unannotated text corpus in the unsupervised way without outer knowledge. We want to show the word associations are useful in IR. We have been mainly used corpora of newspaper articles in **InfoMap** as a source of the general word associations and a target of retrieval [Schütze 97]. We are now interested in how effect to the behavior of retrieval by use of word associations derived from the different training corpora. Thus we decide to do information retrieval experiments based on domain-specific word associations derived from domain-specific corpora.

We use OHSUMED [Hersh 94] (348,566 documents), a part of MEDLINE corpus in medical field (5 year portions of MEDLINE from 1987to 1991) which edited for evaluation for IR systems. In OHSUMED, 106 queries and related documents information (judged by specialists) for the queries are attached. OHSUMED corpus has 2 types of answers, "definitely relevant" (DR : documents fully relevant for queries) and "possibly relevant" (D+PR : "plausibly" relevant for queries) . We think **InfoMap** is suitable for retrieval of latent related documents (D+PR). Unfortunately, [Hersh 94] only describes the results for DR by Cornell's SMART system. So we cannot compare about D+PR directly. Then, we tried to evaluate about DR. Table. 2 shows a part of the experimental result.

Average recall for OHSUMED by SMART system

TABLE 2. Experiment result for OHSUMED (recall)

		Recall %		
Smart	# of docs	5 docs	15docs	100 docs
average	348,566	11.5	21.7	48.8
InfoMap	# of docs	5 docs	15docs	100 docs
(20k words)	54,710	12.5	23.2	51.5
(30k)	54,710	13.8	25.0	53.3
(30k) stem	54,710	14.6	23.8	54.2
(30k)	124,535	7.39	17.4	38.7
(30k) stem	124,535	9.24	16.6	41.8

- () : # of word used in co-occurrence count
- # of content-bearing words are all 1000.
- "stem" means processed by stemming

[2] To find out similar words, the closeness (proximity) of word vectors is also calculated by the cosine measure.

(5 different parameter settings) described in [Hersh 94] are severally 11.5, 21.7, 48.8 (%) for top 5, top 15, and top 100 outputs ranked by similarity with queries. Evaluation result in InfoMap for 54,710 documents, 1987 portions of OHSUMED, is recall 12.5, 23.2, 51.5 (%) seems to be good. But for 124,535 documents of 1987–88 portions, Recalls are relatively low. And Recalls becomes lower as increasing the number of documents (We omitted the result for more than 3 years documents in Table 2). We use same corpora for calculation of word vectors and document vectors and for retrieval target documents (close test) in this experiment. We think the deterioration comes from "over fitting".

The size of word count dictionary extracted from all OHSUMED corpus is about 273 k words (228 k words with stemming). We usually use the size of co-occurrence matrix 20,000 by 1,000 for experiments for news articles in InfoMap. We used 30,000 by 1,000 for OHSUMED because of the size of the dictionary. This improves recall as shown in Table 2. "Stem" in Table 2 means with stemming, the others are no stemming. The stemming improves recall in top 5 and top 100 documents. (However, recall for top 15 becomes lower). In this experiment, we did not use the statistical phrases as one words in the co-occurrence matrix described in Section 2.1.4. We expect the use of the statistical phrases improve the retrieval efficiency.

4. CONCLUSION

The text corpus for information retrieval consists of a collection of words, so representation of word senses is an essential basis for IR. We use word vectors derived from text corpora as a representation of word senses. To lower the dimensionality of word vectors in order to enhance their tractability, we use SVD as a dimensionality reduction tool, which is a method of liner algebra. We explained the calculation using SVD is nearly equivalent to PCA. Furthermore we extend word vectors to context vectors and construct IR system called InfoMap. This paper describes how to create word vectors, the configuration of InfoMap system, and an experimental result using a domain-specific corpus. The experiment described here is very preliminary, we are planning to investigate the effect of the use of domain-specific word associations to the behavior of IR. As related studies, we made an experiment for personalized word associations derived from personal e-mails which potentially reflects personal interests [Flournoy 98]. We also tried term-list translation experiment between English and Japanese [Kikui 98] for seeking the applicability of InfoMap techniques to multilingual IR.

ACKNOWLEDGMENTS

We would like to thank Hinrich Schütze and members of Infomap project at Stanford.

REFERENCES

BERRY, M. W. 1992. Large Scale Singular Value Computations, International Journal of Supercomputer Applications, 6(1):13–49.

DEERWESTER, S., S. T. DUMAIS, G. W. FURNAS, T. K. LANDAUER, and R. HARSHMAN. 1990. Indexing by latent semantic analysis. J. American Society for Information Science, 41(6):391–407.

FLOURNOY, R., R. GINSTROM, K. IMAI, S. KAUFMANN, G. KIKUI, S. PETERS, H. SCHÜTZE, Y. TAKAYAMA. 1998. Personalization and Users' Semantic Expectations. ACM SIGIR '98 Post-Conference Workshop on Query Input and User Expectations, 31–35.

HERSH, W. R., C. BUCKLEY, T. J. LEONE, D. H. HICKAM 1994. OHSUMED: An interactive retrieval evaluation and new large test collection for research. ACM SIGIR '94, 192–201.

KIKUI, G. 1998. Term-list Translation using Mono-lingual Word Co-occurrence Vectors, COLING-ACL '98, 670–674.

KOZIMA, H., A. ITO. 1997. A Context-sensitive Measurement of Semantic Word Distance. Journal of IPSJ, **38**(3):482–489 (in Japanese).

LAY, D. C. 1997. Linear Algebra and its applications. revised ed.

NIWA, Y., Y. NITTA. 1994. Co-occurrence Vectors from Corpora vs. Distance Vectors from Dictionaries. COLING '94, 304–309.

PORTER, M. F. 1980. An algorithm for suffix stripping. Program, **14**, 130-137.

SALTON, G., A. WANG, C. S. YANG. 1975. A vector space model for automatic indexing. Comm. ACM, **18**, 613–620.

SCHÜTZE, H. 1997. Ambiguity Resolution in Language Learning: Computational and Cognitive Models. CSLI Lecture Notes **71**, CSLI Publications. (Ph.D. thesis, Stanford, Linguistics, 1995.)

SCHÜTZE, H., J. O. PEDERSEN. 1997. A cooccurrence-based thesaurus and two applications to information retrieval. Information Processing & Management, **33**(3): 307–318.

SCHÜTZE, H. 1998. Automatic Word Sense Discrimination. Computational Linguistics. **24**(1):97–123.