IJCSN International Journal of Computer Science and Network, Volume 4, Issue 3, June 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

A Sentence/Word-Based LRA Using Hybrid Latent Semantic Indexing

[1] Kiran Saroha, [2] Sukhdip Singh

[1, 2,] CSE Department, DeenBandhuChhotu Ram University of Science and Technology,
Murthal, Haryana, India

Abstract - In this Research Paper we propose a description-LRA, also known as Latent Semantic Indexing, an application in information retrieval that promises to offer better performance by overcoming some limitations that plagues traditional term matching techniques. These term matching techniques have always relied on matching query terms with document terms to retrieve the documents having terms matching the query terms.. (LRA) seems to be a promising technique in overcoming the natural language problems between terms and documents that are mapped and closely related to each other. Queries are then mapped to this space with documents being retrieved based on similarity Model.

Keywords - *LSI, NLP, Cholesky Transform, SVM, LUD.*

1. Introduction

LRA (LSI) commonly known as Latent Semantic Indexing in the context of information retrieval is a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. It is based on the application of a particular mathematical technique, called Singular Value Decomposition (SVD), to a word-by-document matrix .The word-by-document matrix is formed from LSI inputs that consist of raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs. This application provides a way of viewing the global relationship between terms in the whole documents' collection enabling the semantic structures within the collection to be unearthed. LSI enables retrieval on the basis of conceptual content, instead of merely matching words between queries and documents. By use of dimensionality reduction which makes complex natural languages problems tractable and the intrinsically global outlook of the approach.

For Latent Semantic Indexing on a group of documents, following steps should be performed:

- First, convert each document in your index into a vector of word occurrences. The number of dimensions your vector exists in is equal to the number of unique words in the entire document set. Most document vectors will have large empty patches, some will be quite full. It is recommended that common words (e.g., "this", "him", "that", "the") are removed.
- Next, scale each vector so that every term reflects the frequency of its occurrence in context. *I'll post the math for this step when I get home.* Seems he didn't ever get home ;-)
- Next, combine these column vectors into a large *term-document matrix.* Rows represent terms, columns represent documents.
- Perform Singular Value Decomposition on the term-document matrix. This will result in three matrices commonly called U, S and V. S is of particular interest, it is a diagonal matrix of singular values for your document system.
- Set all but the *k* highest singular values to 0. *k* is a parameter that needs to be tuned based on your space. Very low values of *k* are of poor results. But very high values of *k* do not change the results much from simple vector search. This makes a new matrix, S'.
- Recombine the terms to form the original matrix (i.e., U * S' * V(t) = M' where (t) signifies transpose).
- Break this reduced rank term-document matrix back into column vectors. Associate these with their corresponding documents.
- At last, Latent Semantic Index has been generated.

2. Literature Survey

The detection of antonym has been studied in a number of previous papers. Mohammed et al. (2008) approach the problem by combining information from a published thesaurus with corpus statistics derived from the Google n-gram corpus (Brants and Franz, 2006). Their method consists of two main steps: first, detecting contrasting word categories (e.g. "WORK" vs. "ACTIVITY FOR FUN") and then determining the degree of antonym. Categories are defined by a thesaurus; contrasting categories are found by using affix rules (e.g., un- & dis-) and WorldNet antonym links. Words belonging to contrasting categories are treated as antonyms and the degree of contrast is determined by distributional similarity.Mohammed et al. (2008) also provides a publicly available dataset for detection of antonyms, which we have adopted. This work has been extended in (Mohammed et al., 2011) to include a study of antonym based on crowd-sourcing experimentsTurney (2008) proposes a unified approach to handling analogies, synonyms, antonyms and associations by transforming the last three cases into cases of analogy. A supervised learning method is then used to solve the resulting analogical problems.

3. Tables, Figures and Equations

3.1 Tables and Figures

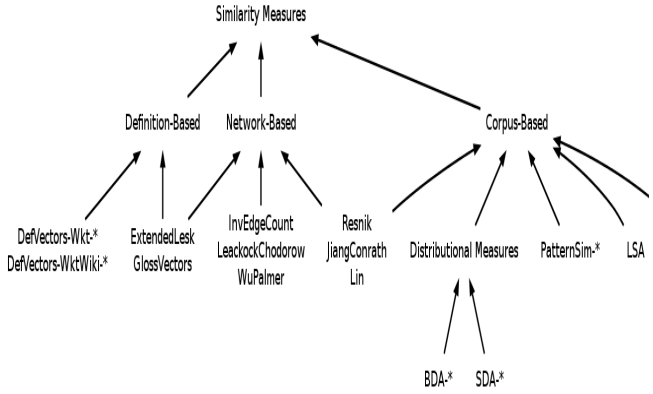

Fig 1:Classification of the semantic similarity Model used compared in this section

3.2 Equations

3.2.1 LU Decomposition

If *A* is a square matrix and it can be factored as $\boldsymbol{A = LU}$ where *L* is a lower triangular matrix and *U* is an upper triangular matrix, then we say that *A* has an LU-Decomposition of *LU*.

If A is a square matrix and it can be reduced to a row-echelon form, U, without interchanging any rows , then A can be factored as $\boldsymbol{A = LU}$ where *L* is a lower triangular matrix.

LU decomposition of a matrix is not unique.

There are three factorization methods:
Crout Method: dig (U) = 1; $\boldsymbol{u_{ii} = 1}$
Doolittle Method: dig (L) = 1; $\boldsymbol{l_{ii} = 1}$
Cholesky Method: dig (U) = dig (L) ;$\boldsymbol{u_{ii} = l_{ii}}$

To solve several linear systems $\boldsymbol{Ax = b}$ with the same A, and A is big, we would like to avoid repeating the steps of Gaussian elimination on A for every different B. The most efficient and accurate way is LU-decomposition, which in effect records the steps of Gaussian elimination. This is Doolittle Method.

Without pivoting:
$\boldsymbol{Ax = b}$
$\boldsymbol{LUx = b}$

To solve this, first we solve $\boldsymbol{Ly = b}$ for $\boldsymbol{y}$by forward-substitution method,
then solve $\boldsymbol{Ux = y}$ for $\boldsymbol{x}$ by backward-substitution method.

With pivoting:
$\boldsymbol{Ax = b}$
$\boldsymbol{PAx = Pb}$, where P is permutation matrix.
$\boldsymbol{LUx = Pb}$
To solve this, first we solve $\boldsymbol{Ly = Pb}$ for $\boldsymbol{y}$by forward-substitution method,

then solve $\boldsymbol{Ux = y}$ for $\boldsymbol{x}$ by backward-substitution method.

The main idea of the LU decomposition is to record the steps used in Gaussian elimination on A in the places where the zero is produced.
Let's see an example of LU-Decomposition without pivoting:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & \mathbf{-2} & \mathbf{3} \\ \mathbf{2} & \mathbf{-5} & \mathbf{12} \\ \mathbf{0} & \mathbf{2} & \mathbf{-10} \end{bmatrix}$$

The first step of Gaussian elimination is to subtract 2 times the first row form the second row. In order to record what was done, the multiplier, 2, into the place it was used to make a zero.

$$\xrightarrow{R2-2R1} \begin{bmatrix} \mathbf{1} & \mathbf{-2} & \mathbf{3} \\ (\mathbf{2}) & \mathbf{-1} & \mathbf{6} \\ \mathbf{0} & \mathbf{2} & \mathbf{-10} \end{bmatrix}$$

There is already a zero in the lower left corner, so we don't need to eliminate anything there. We record this fact with a (0). To eliminate a_{32}, we need to subtract -2 times the second row from the third row. Recording the -2:

$$\xrightarrow{R3-(-2)R2} \begin{bmatrix} 1 & -2 & 3 \\ (2) & -1 & 6 \\ (0) & (-2) & 2 \end{bmatrix}$$

Let U be the upper triangular matrix produced, and let L be the lower triangular matrix with the records and ones on the diagonal:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & -2 & 3 \\ 0 & -1 & 6 \\ 0 & 0 & -10 \end{bmatrix}$$

Then,

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 3 \\ 0 & -1 & 6 \\ 0 & 0 & -10 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -2 & 3 \\ 2 & -5 & 12 \\ 0 & 2 & -10 \end{bmatrix} = A$$

Go back to the first example, rewrite the system of equation into matrix equation:

$$\begin{bmatrix} 1 & -3 & 1 \\ 2 & -8 & 8 \\ -6 & 3 & -15 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 9 \end{bmatrix} Ax = b$$

$$A: \begin{bmatrix} 1 & -3 & 1 \\ 2 & -8 & 8 \\ -6 & 3 & -15 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & -3 & 1 \\ (2) & -2 & 6 \\ (-6) & -15 & -9 \end{bmatrix}$$
$$\rightarrow \begin{bmatrix} 1 & -3 & 1 \\ (2) & -2 & 6 \\ (-6) & \left(-\frac{15}{2}\right) & -54 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -6 & -\frac{15}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 & 1 \\ 0 & -2 & 6 \\ 0 & 0 & -54 \end{bmatrix}$$
$$= LULU - Decomposition$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -6 & -\frac{15}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 & 1 \\ 0 & -2 & 6 \\ 0 & 0 & -54 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 9 \end{bmatrix} LUx = b$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -6 & -\frac{15}{2} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \\ 9 \end{bmatrix} sloveLy$$
$$= busingforwardsubstitution$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -10 \\ 108 \end{bmatrix}$$
$$= \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$
$$\begin{bmatrix} 1 & -3 & 1 \\ 0 & -2 & 6 \\ 0 & 0 & -54 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
$$= \begin{bmatrix} 4 \\ -10 \\ 108 \end{bmatrix} \quad solve\ Ux$$
$$= y\ using\ backward\ substitution$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ -2 \end{bmatrix}$$

3.2.2 NLP (Natural Language Processing)

- Reading in the meetings transcripts
- Parsing by speech
- (stop words, stemming etc. are options – benefits less obvious for text reuse)
- Writing a swalign function
- Calling that function and applying it to the file list(s)
- Investigating whether you learn anything from the speeches that produce the highest similarity scores

4. Conclusions

LRA has developed an interesting alternative to the current Window search tools. It concerns the development of particular s able to seek out and collect subsets of Window pages related to given database. Many domains may benefit from this research, such as vertical portals and personalized search systems, which provide interesting information to communities of users. Some of the most interesting approaches have been described, along with important algorithms, such as LSI, SVD, NLP, and LUD. Particular attention has been given to adaptive learnable s, where learning methods are able to adapt the system behavior to a particular environment and input parameters during the search.

5. Future Work

The chance to activity such data in the process could help retrieving information more quickly, reducing the network and computational resources. The user-interest LRA construction is proposed by using user log profile. Based

on user-interest LRA, we proposed the seed URLs selection approach. In future it may be developed in other domains.

References

[1] C. A. Bechikh and Haddad, H., A quality study of noun phrases as document keywords for information retrieval, International Conference on Control, Engineering and Information Technology, 2013

[2] Ceri Stefano et al, Web Information Retrieval. Berlin, Springer, 2013

[3] Evans D.A. and Zhai, C., Noun-Phrase Analysis in Unrestricted Text for Information Retrieval, *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 1996. .

[4] Grossman David A. and Frieder, Ophir, *Information retrieval: algorithms and heuristics, 2nd Edition*. Chicago, Springer, 2004

[5] Lahtinen T, Automatic indexing: an approach using an index term corpus and combining linguistic and statistical methods, PhD thesis, University of Helsinki, 2000.

[6] Manning Christopher D., Raghavan, Prabhakar and Schütze, Hinrich, An Introduction to Information RetrievaCambridge, England, Cambridge University Press,2000

[7] J Cho and H. Garcia-Molina, Synchronizing a database to improve freshness SIGMOD Record 29, 2, 2000, pp. 117–128.

[8] Chakrabarti, Soumen, Martin van den Berg, and Byron Dom., Focused: a new approach to topicspecific Web resource discovery, Elsevier, 1999.. .

[9] Rajender Nath and Satinder Bal, A Novel Mobile System Based on Filtering off Non-Modified Pages for Reducing Load on the Network, published in The International Arab Journal of Information Technology, Vol. 8, No. 3, July 2011.

[10] Rajender Nath and Satinder Bal, Reduction in Bandwidth Usage for Using Mobile s, published in International Journal of Computer Science and Knowledge Engineering, January-December 2007, pp. 51-61, ISSN: 0973-6735..