# NLP Project : Non-Orthogonal Explicit Semantic Analysis

Akshay Utture, Anirudh Alumeluvari, Kshitij Tomar, and Nikhil Maryala

Indian Institute of Technology, Madras

**Abstract.** The traditional techniques for computing word or document similarity are Latent Semantic Analysis, Distributional Similarity, and WordNet based Similarity. Recently the most effective technique has been Explicit Semantic Analysis (ESA). ESA represents every word or document in the vector space represented by Wikipedia articles. In this technique, all Wikipedia articles are assumed orthogonal. To overcome this, there Non-Orthogonal ESA was introduced, where this orthogonality assumption is dropped. In this project, we explore a few new ideas in the Non-Orthogonal ESA technique. For the evaluation of our new ideas, we use this representation of documents in terms of Wikipedia articles for Text Classification.

**Keywords:** Natural Language Processing, Explicit Semantic Analysis, Document Similarity

## 1 Introduction

### 1.1 Explicit Semantic Analysis

The standard similarity measures for calcualting word or document overlap, like cosine similarity with tf-idf or word distributional similarity, try to use co-occuring words or words occuring in the same document to compute similarity. These methods do not try to look at the semantics of words. Latent Semantic Analysis [3](Scott Deerwester, Susan Dumais, George Furnas, Richard Harshman, Thomas Landauer,1998) tried to capture the concept of semantics by represnting words in a latent semantic space using Singular Value Decomposition. Explicit Semantic Analysis [2](Evgeniy Gabrilovich and Shaul Markovitch, 2007) tries to capute this notion of semantics using a huge source of world knowledge - Wikipedia, that was handwritten by humans. Words and documents are now expressed as vectors, according to their relatedness to these Wikipedia articles which are treated as concepts.
Explicit Semantic Analysis achieved better results than any of the previous methods, and also resulted in interpretability because we could pick up the top 10 concepts that a word or document was related to by taking the components with the highest value. The example that the authors of [2] give for the same is shown in table 1.

**Table 1.** Top ten related Wikipedia articles for some words

| **Input: equipment** | **Input: investor** |
|---|---|
| Tool | Investment |
| Digital Equipment Corporation | Angel investor |
| Military technology and equipment | Stock trader |
| Camping | Mutual fund |
| Engineering vehicle | Margin (finance) |
| Weapon | Modern portfolio theory |
| Original equipment manufacturer | Equity investment |
| French Army | Exchange-traded fund |
| Electronic test equipment | Hedge fund |
| Distance Measuring Equipmen | Ponzi scheme |

To compute the Wikipedia Vector for a word, we just use the frequency of the word along each Wikipedia Article (as a ratio to the total number of words in the wikipedia article) to calculate the component of the word along that Wikipedia Article. To compute the Wikipedia Vector for a document we simply add the Wikipedia vectors of all its words 1 [1]

The authors used Spearman Rank-Order Correlation to show that this method performed significantly better that other methods tried up to that point.

So if the Wikipedia vectors of 2 words $w1$ and $w2$ is $\overrightarrow{w1}$ and $\overrightarrow{w2}$ respectively, then word similarity can be computed using the cosine similarity measure. (as given by [2])

$$Similarity(w1, w2) = \frac{\overrightarrow{w1}.\overrightarrow{w2}}{|\overrightarrow{w1}|.|\overrightarrow{w2}|} \tag{1}$$

## 1.2 Why Non-Orthogonal ESA

ESA assumes that all the Wikipedia articles are orthogonal to each other, which is in fact not true. For example, the articles on 'Iraq' and 'Iraq War' are not orthogonal. Words occurring in the 'Iraq War' article, but not in the 'Iraq' article should also have some component along the Iraq article for the correct semantic representation. An example given by [1] to show this problem is given in Table 2. It shows the top 5 articles for the words 'football' and 'soccer' are completely different. These words actually have a very high similarity, but if we take the cosine similarity between the 2 words like in equation (1), they will score very less. Even though the top 5 articles are not the same in both, they are very related.

[1] It should be noted that all stop words are removed, punctuation and orthographic features are dropped.

The Non-Orthogonal ESA tries to capture this relatedness between Wikipedia articles to give a better semantic representation for each word. When this better semantic representation is used for each word, it can give better results when used for computing word similarity, document similarity, feature extraction for text classification,etc.

**Table 2.** Why ESA can fail : Top 5 related Wikipedia articles for football and soccer

| **Input: football** | **Input: soccer** |
|---|---|
| FIFA | History of soccer in the United States |
| Football Equipment Corporation | Soccer in the United States |
| History of association football | United States Soccer Federation |
| Football in England | North American Soccer League (196884) |
| Association football | United Soccer Leagues |

### 1.3 Organization of the rest of the Project Report

Section 2 talks about various Non-orthogonal ESA (NESA) Techniques, including the methods proposed by Nitish Aggarwal,Kartik Asooja, Georgeta Bordea and Paul Buitelaar in their paper - [1], as well as some new ideas proposed by us. Section 3 describes the results we obtained when we tried to evaluate these NESA techniques by doing feature extraction on text, and then text classification using these features. Section 4 gives the possible improvements that could have been made in the implementation. Section 5 concludes the report.

## 2 Non-orthogonal ESA

In order to drop the orthogonality assumption in ESA, we need to find some way to define the similarity between Wikipedia articles, and use this to refine our measure of similarity between words, or the semantic representation of words and documents.

### 2.1 General Framework for Non-Orthogonal ESA methods

We define a matrix $C$, which is called the correlation matrix. If $N$ in the number of Wikipedia we are using, then $NxN$ are the dimensions of $C$. The entry $C[i][j]$ represents the similarity between the 'i$^{\text{th}}$' and 'j$^{\text{th}}$' Wikipedia article. If the similarity measure is symmetric, which is the case in all the methods described below, $C$ will be a symmetric matrix. If not, then it need not be symmetric. Now, to compute the similarity between 2 words, we use do the following. If the

$N$-dimensional normalized Wikipedia vectors of 2 words $w1$ and $w2$ is $\overrightarrow{w1}$ and $\overrightarrow{w2}$ respectively, then to calculate similarity we use this equation. (as given by [1])

$$Similarity(w1, w2) = \overrightarrow{w1}.C.\overrightarrow{w2} \tag{2}$$

The authors of [1] additionally define a Matrix $E$, called the Transformation Matrix, which is an $NxM$ matrix that is used to transform a Wikipedia vector to an $M$ dimensional vector space. The word similarity is then computed in this new vector space. (as given by [2])

$$Similarity(w1, w2) = \frac{(\overrightarrow{w1}.E).(\overrightarrow{w2}.E)}{|(\overrightarrow{w1}.E)|.|(\overrightarrow{w2}.E)|} \tag{3}$$

matrices $C$ and $E$ are related in the following way

$$C = E^T.E \tag{4}$$

In case we want to find the new semantic representation of a word in terms of Wikipedia concepts, after dropping the assumption of Non-Orthogonality, the representation of the word $w1$ is $\overrightarrow{w1}.C$

If we want to use Non-Orthogonal ESA for feature extraction from text and then do text classification, then for each document in the training set, we treat the document as a bag of words, and for every word, we compute its NESA Wikipedia vector (ie. $\overrightarrow{w1}.C$), and the document's NESA Wikipedia vector is the sum of the vectors of its words. This NESA Wikipedia vector is taken as the set of features for that text. After this step, any standard classifer can be trained on these set of features, and used for text classification.

In the methods given below, we explore different ways of computing the $E$ matrix or the $C$ matrix.

### 2.2 Methods used in the paper [1] to Compute $C$

The paper [2] talks about 4 methods for computing the $C$ matrix, namely, VSM-Text, VSM-Hyperlink, ESA-WikiTitle, DiSER.

**VSM-Text :** This method uses the simple Vector Space model to compute similarity. Each Wikipedia article is represented as a vector of the words occurring in it. The component along a word is equal to the tf-idf (term frequency - inverse document frequency) value of the word. Similarity between 2 articles $i$ and $j$ or in other words, $C[i][j]$ is computed using cosine similarity or using the $E$ matrix. This is basically a word overlap based similarity measure. The $E$ matrix here is a matrix of$N$ Wikipedia articles along the rows, and $M$ words2 [2] along the columns. $E[i][j]$ gives the tf-idf value of word $j$ in Wikipedia article $i$.

[2] Again, all stop words are removed, punctuation and orthographic features are dropped.

**VSM-Hyperlink :** This method also uses the simple Vector Space model to compute similarity. But in this case, each Wikipedia article is represented as a vector of the hyperlinks occurring in it. The component along a hyperlink is equal to the tf-idf (term frequency - inverse document frequency) value of the hyperlink. Similarity between 2 articles $i$ and $j$ or in other words, $C[i][j]$ is computed using cosine similarity or using the $E$ matrix. This is basically a word overlap based similarity measure. The $E$ matrix here is a matrix of$N$ Wikipedia articles along the rows, and $M$ hyperlinks 2 along the columns. $E[i][j]$ gives the tf-idf value of hyperlink $j$ in Wikipedia article $i$. It should be noted that the $E$ matrix here will be sparse and a sparse representation will help.

**ESA-WikiTitle :** In this method, we use ESA itself to compute the relatedness between Wikipedia concepts. Each Wikipedia article has a title associated with it, and this title is represented using ESA in the Wikipedia concept space. So the title is treated as a document consisting of the words in the title. The ESA representation of the title is done as explained in section (1.1). So the $E$ matrix can be understood like this. The row $i$ is the ESA representation of the title of the Wikipedia article number $i$. For this method, the $E$ matrix has $N = M$ since the columns also represent Wikipedia articles.

## 2.3 Some new methods proposed by us to Compute *C*

We have suggested four new methods for computing the $C$ matrix - NESA-LSA, ESA-WikiBody, NESA-LinearCombination, NESA - Category Tree Distance Based Measure

**NESA-LSA :** In this method, we treat each Wikipedia article as a document, and compute the Word-Document matrix, which represents the tf-idf value for every word in every document. The rows represent different Wikipedia articles, and the columns represent the distinct words appearing in theses Wikipedia articles. We implement the standard Latent Semantic Analysis technique as mentioned in paper [3]. We will compute the Singular Value Decomposition of this Word-Document Matrix. Now the document-document similarity can be computed just as it is in [3].i.e. we represent every document by a vector in the latent semantic space and similarity is computed by cosine similarity measure. The only problem with this method is that it is a little computationally expensive. If the number of Wikipedia articles exceeds 100,000 this method may take too much time. The time complexity of the best algorithms for SVD computation of a $mxn$ Word-Document matrix is $O(m^2 * n + n^3)$ for an algorithm called R-SVD.

**ESA-WikiBody :** This method is very similar to ESA-WikiTitle and is derived from it. Instead of using just the title for computing the Wikipedia Vector for a Wikipedia article, we use the entire text of the Wikipedia article, without stop words. Each Wikipedia article is represented using ESA in the Wikipedia concept space. So the Wikipedia article is treated as a document consisting of the words. The ESA representation of the document is done as explained in section (1.1). So the $E$ matrix can be understood like this. The row $i$ is the ESA representation of the Wikipedia article number $i$. For this method, the $E$ matrix has $N = M$ since the columns also represent Wikipedia articles.

**NESA-LinearCombination :** This method proposes that perhaps no 1 measure is good enough for computing similarity, but a linear combination of them may prove to be better. For the similarity between any 2 Wikipedia articles $i$ and $j$ ($C[i][j]$), we take the similarity computed by all the 4 measures in section 2.2, and combine them in a weighted fashion. The drawback of this method is that we need a way to effectively compute the weights, or find a way to learn the weights. Since we could not come up with a satisfactory way to do either of these things, we did not pursue this idea further.

**NESA - Category Tree Distance Based Measure :** WordNet can be used to compute similarity between nouns, as explained in the paper [4]. One of the techniques is to use the WordNet noun taxonomy, and compute the similarity to be a function of the inverse of the distance between the pair of nouns in the taxonomy. Optionally, we can also make the similarty of function of the Lowest Common Ancestor of the 2 words, in addition to the distance between them. We can use a similar idea for Wikipedia articles. Wikipedia has a Category tree, which is similar to the WordNet noun taxonomy. We can calculate the similarity between Wikipedia articles as a function of the inverse of the distances between Wikipedia articles in this Category tree. The only difference between the WordNet taxonomy and the Wikipedia Category Tree is that a Wikipedia article might occur at multiple places in the category tree. That is, it may be a part of multiple categories. For example, the Wikipedia article on Statistics in the following 5 categories : Statistics, Data, Formal Sciences, Information, Mathematical and quantitative methods (economics) Hence we can say that the distance between 2 categories is the minimum distance between the 2 articles. If the 2 articles are $a1$ and $a2$

$$Similarity(a1, a2) = f(dist(a1, a2), LCA(a1, a2)) \tag{5}$$

where $dist(a1, a2)$ is the minimum distance between $a1$ and $a2$ in the Wikipedia Category Tree, and $LCA(a1, a2)$ is the depth of the Least Common Ancestor in the shortest distance path.

## 3 Implementation and Results

### 3.1 Evaluation Measure for Each Method

In order to evaluate how well each method performed, we used text classification. ESA or NESA was used for feature extraction on the training and test input in

order to get a set of features. A standard classification was then performed on these extracted features.
The reason why text classification was used for evaluation instead of other methods like the Spearman Rank Correlation for Human judged values because the data sets available are small, prone to subjectivity, and not commonly used. Text classification gives an objective way of testing the effectiveness of the new measures. The higher the accuracy, the better the measure.

### 3.2 Datasets used

A set of 4531 Wikipedia articles were used as the concepts. The number of articles picked is so few because anything larger will take several hours to run. Even with these many articles, the current implementation took around 20 minutes to run.

| **Classes in the 20 Newsgroup Dataset** |
|---|
| comp.graphics |
| comp.os.ms-windows.misc |
| comp.sys.ibm.pc.hardware |
| comp.sys.mac.hardware |
| comp.windows.x |
| rec.autos |
| rec.motorcycles |
| rec.sport.baseball |
| rec.sport.hockey |
| sci.crypt |
| sci.electronics |
| sci.med |
| sci.space |
| comp.sys.mac.hardware |
| comp.windows.x |
| rec.autos |
| misc.forsale |
| talk.politics.misc |
| alt.atheism |
| soc.religion.christian |

The 20 NewsGroup dataset (http://qwone.com/ jason/20Newsgroups/) was used for the purpose of text classification. The Wikipedia articles that were chosen were picked roughly from the set of categories in the NewsGroup dataset. This dataset has around 20,000 text articles which are approximately split as 60 percent train and 40 percent test. This is a popular data set, and has been used in [5] for the purpose of Text Classification using ESA

The data set has 20 classes which are listed out in Table 3. It should be noted that some of the classes are similar and hence will be close together in their semantic representation (like MAC hardware and IBM PC hardware) , where as some classes are fat apart (like Atheism and Motorcycles).

### 3.3 Results

The code for each of the NESA methods, and ESA was implemented in Java. The 'JAMA' Java library was used for the Singular Value Decomposition in the NESA-LSA method. After feature extraction, the classification task was performed in Python using the Scikit-Learn library. The classifier used was the Random Forest Classifier. The reason this was picked was that it is a powerful classifier, which works well with minimal parameter tuning. Since each method was taking quite long to run, parameter estimation with cross validation could have take several hours for each method.

The methods that were implemented were ESA as given in paper [2]. From paper [1], VSM-Text, VSM-Hyperlink, ESA-WikiTitle was implemented. Among the new methods proposed by us, NESA-LSA and ESA-WikiBody, was implemented. The reason for not implementing the Linear Combination in given in section (2.3). The reason the NESA - Category Tree Distance Based Measure was not implemented was that we needed a tree of sufficient depth to get any kind of benefits from this method. But since our implementations were not efficient enough, we could not use more than 5000 articles, and this would result in a very short category tree : probably of 2-3 levels at most.

The accuracy scores for each of the implemented methods for text classification on the 20 News Group data set is given in Table 4.

**Table 3.** Accuracy score for Text Classification

| **Method** | **Accuracy (in percentage)** |
|---|---|
| ESA | 60.2 |
| VSM-Text | 61.6 |
| VSM-Hyperlink | 63.6 |
| ESA-WikiTitle | 63.1 |
| NESA-LSA | 63.0 |
| ESA-WikiBody | 62.3 |

## 4 Possible Improvements in Code Implementation

*Inefficient Implementation of Data Structures* The implementation of the data structures required was not very efficient or optimal. Array representations were used in some places, and sparse representation was used in others. Memory started running out in places that we had used the array representation. So a sparse representation should have been used in all places.

*Using more Wikipedia articles* Only 4531 Wikipedia articles were used for this implementation. We started out with close to 10,000. But the code took very long to run, and we had to prune it down to 4531.

*Computing On the Fly* Perhaps the reason why we ran out of memory could have been because we pre-computed everything. If we computed some of the data structures on the fly, we might have been able to accommodate several more Wikipedia articles.

*Handpicking articles carefully* Articles were chosen that roughly pertained to the topic. Smaller articles were pruned out. We could have handpicked articles more carefully, except that it would have been a time consuming task. Alternatively, we could have used some standard Wikipedia article collections like Yago or DBPedia, but both of these have over 100,000 articles in their collection, and we did not have an efficient implementation to handle even 10000.

## 5 Conclusion

In conclusion, dropping the orthogonality assumption in ESA is useful. This can be concluded because all the Non-orthogonal ESA based methods outperform ESA in the text classification task. The overall accuracy of the implemented methods is still fairly low. More efficient implementations of the data structures will allow us to include more number of Wikipedia articles and give better answers, with

## References

1. Nitish Aggarwal, Kartik Asooja, Georgeta Bordea, Paul Buitelaar; Non-Orthogonal Explicit Semantic Analysis (2014)
2. Evgeniy Gabrilovich and Shaul Markovitch; Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis (2007)
3. Scott Deerwester, Susan T. Dumais, George W. Furnas, and Thomas K. Landauer, Richard Harshman ; Indexing by Latent Semantic Analysis (1908)
4. George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller; Introduction to WordNet: An On-line Lexical Database (1993)
5. Ming-Wei Chang, Lev Ratinov, Dan Roth and Vivek Srikumar; Importance of Semantic Representation: Dataless Classification; Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)