Text classification using document-document semantic similarity

Indrajit Mukherjee*

Department of Computer Science and Engineering,
BIT Mesra, Ranchi, India
E-mail: imukherjee@bitmesra.ac.in
*Corresponding author

Prabhat Kumar Mahanti

Department of Computer Science and Applied Statistics (CSAS),
University of New Brunswick Canada, Canada
E-mail: pkmahanti@yahoo.co.in

Vandana Bhattacharya

Department of Computer Science and Engineering,
BIT Mesra, Ranchi, India
E-mail: vbhattacharya@bitmesra.ac.in

Samudra Banerjee

Server Technologies Division,
Oracle India Pvt. Ltd.,
Prestige Lexington Towers, Bangalore, India
E-mail: sam8dec@gmail.com

Abstract: One of the key problems encountered while using a text classification learning algorithms is that they require huge amount of labelled examples to learn accurately. The objective of this paper is to propose a novel method of topic modelling and document-document semantic similarity algorithm (DDSSA), which reduces the need for larger training data. This algorithm finds the concepts and keywords of the unlabelled text, identifying the topic of unlabelled text from list of concepts and keywords obtained from labelled text. This can be achieved by obtaining the concepts of the labelled text and identify the keywords which holds strong relationships with given labelled data. This topics and keywords obtained from the labelled text can be stored in the database which in turn can be used to compute the semantic similarity with concepts obtained from the unlabelled text. The proposed method is compared with the popular latent semantic analysis (LSA) applied in NLTK and Mallet datasets. The experiment result shows that the proposed method is superior to LSA in most of the cases.

Keywords: topic modelling; WorldNet; latent Dirichlet allocation; LDA; latent semantic analysis; LSA.

Reference to this paper should be made as follows: Mukherjee, I., Mahanti, P.K., Bhattacharya, V. and Banerjee, S. (2013) 'Text classification using document-document semantic similarity', *Int. J. Web Science*, Vol. 2, Nos. 1/2, pp.1–26.

Biographical notes: Indrajit Mukherjee is an Assistant Professor in the Department of Computer Science and Engineering, BIT Mesra, Ranchi, India. He obtained his MSc in Electronics from RU, India, MCA degree from BIT Mesra, India and MPhil in Computer Science from AU, India. Currently, he has published papers in various national, international conferences and journals. His research interest includes web information retrieval, text mining, cloud computing, web service applications, and soft computing.

Prabhat Kumar Mahanti is a Professor at the Department of Computer Science and Applied Statistics (CSAS), University of New Brunswick Canada. He obtained his MSc in IIT-Kharagpur, India, and PhD in IIT-Bombay, India. His research interests include software engineering, software metrics, reliability modelling, modelling and simulation, numerical algorithms, finite elements, mobile and soft computing, verification of embedded software, neural computing, data mining, and multi-agent systems. He has more than 300 research papers and technical reports to his credit. He is also involved with several consultancy projects in and around North America. He has produced several PhD students.

Vandana Bhattacharya is a Professor at the Department of Computer Science and Engineering, BIT Mesra, Ranchi, India. She obtained her MTech and PhD in JNU India. Her research interests include software engineering, software metrics, reliability modelling, modelling and simulation, numerical algorithms, finite elements, mobile and soft computing, verification of embedded software, neural computing, data mining, and multi-agent systems. She has more than 200 research papers and technical reports to her credit. She is also involved with several consultancy projects in India. She has produced several PhD students.

Samudra Banerjee has graduated in Computer Science and Engineering in 2010 from Birla Institute of Technology, Mesra, India. He is currently working for Server Technologies Division, Oracle India Pvt. Ltd. He has publications in one national conference, three international conferences and two international journals to his credit. His research interests include database management systems, web information retrieval, text mining and cloud computing.

1 Introduction

Text classification is the task of assigning a text document to a relevant category or categories. Formally, let $C = \{c_1, \ldots, c_K\}$ be a set of predefined categories, $D = \{d_1, \ldots, d_N\}$ be a set of text document to be classified. The task of text document classification is then transformed to approximate the unknown assignment function f, which maps $D \times C$ to a set of real numbers. Each number in the set is a measure of the similarity between a document and a category. Based on the measures, a document is assigned to the most relevant categories (Pantelm and Lin, 2002).

Document representation is one of the most important issues in text classification. In order to be classified, each document should be turned into a machine comprehendible format. The *bag-of-words* document representation (Koller and Sahami, 1998) is simple, yet limited. Attempts have been conducted to improve the effectiveness of the representation. For example, Mladenic (1998) extends the ‘bag-of-words’ to the ‘bag-of-phrases’ and showed improvement of the classification results (Chan, 1999). There are two major problems with the bag-of-words or the bag-of phrases representations. First, it counts word occurrences and omits the fact that a word may have different meanings (or senses) in different documents or even in the same document. For example, the word ‘bank’ may have at least two different senses, as in the ‘Bank’ of India or the ‘bank’ of Ganga River. However, counting word occurrences, these two instances of ‘bank’ are treated as a same feature. The second major problem lies in the fact that sometime related documents may not share the same keywords so that two related documents cannot be recognised as belonging to the same category. Thus, rather than counting word occurrences, counting word senses might improve text classification. Sense-based text classifications (Scott and Matwin, 1998) are attempts to address the problems. These drawbacks are overcome by our proposed algorithm to identify the similar nearly similar documents as well as distinguishing between dissimilar documents in a much better manner.

Measuring the similarity between documents and queries has been extensively studied in information retrieval. These tasks include query reformulation, sponsored search and image retrieval. Standard text similarity measures perform poorly on such tasks because of data sparseness and the lack of context. Measuring the similarity between documents and queries has been extensively studied in information retrieval. However, there are a growing number of tasks that require computing the similarity between two very short segments of text (Metzler et al., 2007). Retrieving documents in response to a user query is the most common text retrieval task. For this reason, most of the text similarity measures that have been developed take as input a query and retrieve matching documents. However, a growing number of tasks, especially those related to web search technologies, rely on accurately computing the similarity between two very short segments of text. Example tasks include query reformulation (query-query similarity), sponsored search (query-keyword similarity), and image retrieval (query-image caption similarity), web search-based document retrieval (query-document similarity).

If the query and document do not have any terms in common, then they receive a very low similarity score, regardless of how topically related they actually are. This is well known as the vocabulary mismatch problem. This problem is only exacerbated if we attempt to use these measures to compute the similarity of two short segments of text. For example, ‘USA’ and ‘United States of America’ are semantically equivalent, yet share no terms in common.

Document representation is one of the most important issues in text classification. In order to be classified, each document should be turned into a machine comprehendible format. The *bag-of-words* document representation (Koller and Sahami, 1998) is simple, yet limited. Attempts have been conducted to improve the effectiveness of the representation. For example, Mladenic (1998) extends the ‘bag-of-words’ to the ‘bag-of-phrases’ and showed improvement of the classification results (Chan, 1999). There are two major problems with the bag-of-words or the bag-of-phrases representations. First, it counts word occurrences and omits the fact that a word may have different meanings (or senses) in different documents or even in the same document. For example, the word ‘bank’ may have at least two different senses, as in the ‘Bank’ of India or the ‘bank’ of

Ganga river. However, counting word occurrences, these two instances of 'bank' are treated as a same feature. The second major problem lies in the fact that sometime related documents may not share the same keywords so that two related documents cannot be recognised as belonging to the same category. Thus, rather than counting word occurrences, counting word senses might improve text classification. Sense-based text classifications (Scott and Matwin, 1998) are attempts to address the problems.

The proposed text classification method is based on a novel document-document semantic similarity which is obtained by expansion of query for all the possible senses. The proposed technique is used for finding the semantic distance measures to compute semantic similarity for identifying topic of unlabelled text from set of trained labelled data using topic modelling (LDA). We apply this algorithm to NLTK datasets and Mallet datasets. The effectiveness of our documents classification method is evaluated by comparing the classification results with LSA. The experiment result shows that proposed algorithm is superior to LSA in most of the cases.

The rest of this paper is organised as follows: In Section 2, some of the existing techniques related to vocabulary mismatch problems are discussed. It also focuses on different similarity measures like query-query similarity, query-document similarity. Section 3 discuss about the different similarity measures used for computing similarity between two queries, two documents, query and document. We also describe about the latent semantic analysis (LSA) a widely used technique for computing document-document similarity based on singular value decomposition (SVD). Section 4 presents the proposed document-document semantic similarity algorithm (DDSSA). Different concepts obtained by using topic modelling obtained from sampling techniques based on latent Dirichlet allocation (LDA) are also discussed in this section. In Section 5, details of experimental results are presented to demonstrate the limitations of LSA and how our proposed algorithm overcomes the drawbacks of LSA. Finally, we conclude and discuss future work in the last section.

2 Related work

Many techniques have been proposed to overcome the vocabulary mismatch problem, including stemming, LSA, translation models, and query expansion (April and Pottenger, 2006). Here, we focus on document-document similarity task, where we compare the concepts obtained from a document. Translation models, in a monolingual setting, have been used for the document retrieval, question answering, and detecting text reuse. The goal is to measure the likelihood that some candidate document or sentence is translation (or transformation) of the query.

However, such models are less likely effective on short segments of texts, such as queries, due to the difficulty involved in estimating reliable translation probabilities for such pieces of text (Imran and Sharan, 2009).

Query expansion is a common technique used to convert an initial, typically short, query into a richer representation of the information need (Bhogal et al., 2007; Qiu and Frei, 1993; Fonseca et al., 2005). This is accomplished by adding terms that are likely to appear in relevant or pseudo relevant documents to the original query representation. With query expansion, the user is guided to formulate queries which enable useful results is obtained. The main aim of query expansion (also known as query augmentation) is to select high-quality expansion terms for a query. This process of adding term scan either be manual, automatic or user-assisted. Manual query expansion relies on user expertise to make decisions on which terms to include in the new query.

Sahami and Heilman (2006) proposed a method of enriching short text representations that can be construed as a form of query expansion. Their proposed method expands short segments of text using web search results. The similarity between two short segments of text can then computed in expanded representation space. The expanded representation and DenseProb similarity measure are similar to this approach. However, we estimate term weights differently and analyse how such expansion approaches compare, in terms of efficiency and effectiveness, to other standard information retrieval measures. The work based on term-based query expansion chooses expansion terms from past user queries directly, rather than using them to construct sets of full text documents from which terms are then selected. The method consists of three phases: ranking the original query against the collection of documents; extracting additional query terms from the highly ranked items; then ranking the new query against the collection. Another suggested method for finding relations between queries and phrases of documents based on query logs. They use the hypothesis that click through information available on search engine logs represents an evidence of relation between queries and documents chosen to be visited by users. This evidence is called cross-reference of documents. Based on this evidence, the authors establish relationships between queries and phrases that occur in the documents chosen. These relationships are then used to expand the initial query or to give query suggestions. This approach can also be used to cluster queries extracted from log files. These clusters are used in question answering systems to find similar queries (Fonseca et al., 2005).

The task of automated text categorisation (TC) is well-known information retrieval and machine learning problem concerned with the ‘labelling of natural language texts with thematic categories’ (Sebastiani, 2002). Hierarchical text classifiers had been used by many research communities as they have the potential of decomposing the classification task into smaller classification problems, thus being potentially faster and more robust than their non-hierarchical counterparts (Wetzker et al., 2007).

Many ideas have emerged over years on how to achieve quality results from Web Classification systems, thus there are different approaches that can be used to a degree such as clustering, Bayesian networks, NNs, DTs, support vector machines (SVMs), etc. (Xhemaliet al., 2009).

3 Some definition

3.1 Similarity measures

A similarity measure can represent the similarity between two documents, two queries or one document and one query (Metzler et al., 2007). It is a function which computes the similarity between a pair of text objects. It is possible to rank the retrieved documents in the order of presumed importance. There are many similarity measures proposed in the literature as best similarity measure does not exist yet. Some of the similarity measures are classified into following categories.

3.1.1 Semantic similarity

Semantic Similarity is a concept whereby a set of documents or terms within term lists are assigned a metric based on the likeness of their meaning/semantic content (Noahet al., 2007). This can be achieved for instance by defining a topological similarity, by using ontologies to define a distance between words (a naive metric for terms arranged as nodes in a directed acyclic graph like a hierarchy would be the minimal distance – in separating edges – between the two term nodes), or using statistical means such as a vector space model to correlate words and textual contexts from a suitable text corpus (co-occurrence). Semantic similarity is a confidence score that reflects the semantic relation between the meanings of two sentences. It is difficult to gain a high accuracy score because the exact semantic meanings are completely understood only in a particular context. Some dictionary-based algorithms to capture the semantic similarity between two sentences, which is heavily based on Wordnet semantic dictionary.

3.1.2 Jaccard similarity

The Jaccard similarity (Mining et al., 2009) is a common index for binary variables. It is defined as the quotient between the intersection and the union of the pair wise compared variables among two objects.

The Jaccard index, also known as the Jaccard similarity coefficient (originally coined coefficient by Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets.

The Jaccard coefficient measures similarity between sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The MinHash min-wise independent permutations locality sensitive hashing scheme may be used to efficiently compute an accurate estimate of the Jaccard similarity coefficient of pairs of sets, where each set is represented by a constant-sized signature derived from the minimum values of a hash function.

The *Jaccard distance*, which measures dissimilarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union:

$$J_\delta(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two documents but are not the shared terms.

3.1.3 WordNet::Similarity

WordNet::Similarity (Parwardhan et al., 2003) is an open source software package developed at the University of Minnesota. It allows the user to measure the semantic similarity or relatedness between a pair of concepts (or word senses), and by extension, between a pair of words. The system provides different measures of similarity and relatedness based on the WordNet lexical database (Fellbaum, 1998). The measures of similarity are based on analysis of the WordNet is a hierarchy. The different measures of similarity are as follows:

- Path similarity:

  Path distance similarity: Return a score denoting how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy. The score is in the range 0 to 1, except in those cases where a path cannot be found (will only be true for verbs as there are many distinct verb taxonomies), in which case none is returned. A score of 1 represents identity, i.e., comparing a sense with itself will return 1.

- WUP similarity:

  Wu-Palmer similarity: Return a score denoting how similar two word senses are, based on the depth of the two senses in the taxonomy and that of their least common subsumer (most specific ancestor node). Previously, the scores computed by this implementation did not always agree with those given by Pedersen's Perl implementation of WordNet similarity. However, with the addition of the simulate root flag (see below), the score for verbs now almost always agree but not always for nouns.

  The LCS does not necessarily feature in the shortest path connecting the two senses, as it is by definition the common ancestor deepest in the taxonomy, not closest to the two senses. Typically, however, it will so feature. Where multiple candidates for the LCS exist, that whose shortest path to the root node is the longest will be selected. Where the LCS has multiple paths to the root, the longer path is used for the purposes of the calculation.

3.1.4 F-measure

In information retrieval contexts, precision and recall (Mining et al., 2009) are defined in terms of a set of *retrieved documents* (e.g., the list of documents produced by a web search engine for a query) and a set of *relevant documents* (e.g., the list of all documents on the internet that are relevant for a certain topic).

Precision

In the field of information retrieval, *precision* is the fraction of retrieved documents that are relevant to the search:

$$\text{Precision} = \frac{\left|\{relevantdocuments\}\right| \cap \left|\{retrievaldocuments\}\right|}{\left|\{retrievaldocuments\}\right|}$$

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called *precision at n* or *P@n*.

For example, for a text search on a set of documents precision is the number of correct results divided by the number of all returned results.

Precision is also used with recall, the percent of *all* relevant documents that is returned by the search. The two measures are sometimes used together in the F1 score (or f-measure) to provide a single measurement for a system.

Note that the meaning and usage of 'precision' in the field of information retrieval differs from the definition of accuracy and precision within other branches of science and technology.

Recall

Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{Recall} = \frac{\left|\{relevantdocuments\}\right| \cap \left|\{retrievaldocuments\}\right|}{\left|\{totalrelevantdocuments\}\right|}$$

For example, for text search on a set of documents recall is the number of correct results divided by the number of results that should have been returned

In binary classification, recall is called sensitivity. So it can be looked at as the probability that a relevant document is retrieved by the query.

It is trivial to achieve recall of 100% by returning all documents in response to any query. Therefore, recall alone is not enough but one needs to measure the number of non-relevant documents also, for example, by computing the precision.

It is possible to interpret precision and recall not as ratios but as probabilities:

- *Precision* is the probability that a (randomly selected) retrieved document is relevant.
- *Recall* is the probability that a (randomly selected) relevant document is retrieved in a search.

Note that the random selection refers to a uniform distribution over the appropriate pool of documents; i.e., by *randomly selected retrieved document*, we mean selecting a document from the set of retrieved documents in a random fashion. The random selection should be such that all documents in the set are equally likely to be selected.

A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$F = 2.\frac{Precision.Recall}{Precision + Recall}$$

This is also known as the F-measure, because recall and precision are evenly weighted.

3.1.5 Term vector similarity

Term vector similarity is also known as cosine similarity (Mining et al., 2009). It is a measure of similarity between two vectors by finding the cosine of the angle between them, often used to compare documents in text mining. In addition, it is used to measure cohesion within clusters in the field of data mining. Given two vectors of attributes, *A* and *B*, the cosine similarity, *θ*, is represented using a dot product and magnitude as

$$\text{Similarity} = \cos(\theta) = \frac{A.B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

3.1.6 Concept similarity

Concept similarity (Mining et al., 2009) is a similarity measure used to calculate the similarity between two ontologically related concepts. It is weighted average measure of syntactic, property, coverage and context similarity.

$$\text{ConceptSim}(c_1, c_2) = \frac{w_1 \times synSim + w_2 \times propSim + w_3 \times cvrgSim + w_4 \times w_4 \times ctxtSim}{w_1 + w_2 + w_3 + w_4}$$

3.2 Latent semantic analysis

It is a technique in natural language processing, in particular in vectorial semantics, of analysing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. In the context of its application to information retrieval, it is sometimes called latent semantic indexing (LSI).

A matrix containing word counts per paragraph (rows are represented by unique words and columns are represented by each paragraph) is constructed from a large piece of text and a mathematical technique called SVD is used to reduce the number of columns while preserving the similarity structure among rows. Words are then compared by taking the cosine of any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words. Here, we discuss some of the features and limitations of LSA as follows:

3.2.1 Properties of LSA

1 First, the documents and words end up being mapped to the same concept space.

2 Second, concept space has vastly fewer dimensions compared to the original matrix.

3 LSA is an inherently global algorithm that looks at trends and patterns from all documents and all words.

3.2.2 Limitations of LSA

1 LSA assumes a Gaussian distribution and Frobenius norm which may not fit all problems.

2 LSA cannot handle polysemy (words with multiple meanings) effectively.

3 LSA depends heavily on SVD which is computationally intensive and hard to update as new documents appear.

4 Proposed work

In recent years, huge amount of information is being added to the web and hence it continues to increase with an explosive speed. But we cannot access information effectively and efficiently unless they are properly well organised and indexed. Suppose user submits a web query which comprises of single or multiple words especially for web image or document search. It becomes even worse when the users' query words may be quite different to the ones used in the documents in describing the same semantics. This problem results in lower precisions and recalls of queries. The user may get an overwhelming but large percent of irrelevant documents in the result set. An effective method for solving the above problem is term-based expansion to obtain the conceptual terms from the documents. This can be achieved by using WordNet-based dictionary.

WordNet, like a standard dictionary, contains the definition of words and their relationships (Gong et al., 2005). But it also differs from a standard dictionary in that, instead of being organised alphabetically, WordNet is organised conceptually. The basic unit in WordNet is a synonym set, or synset, which represents lexicalised concept. For example, the noun 'software' in WordNet 2.0 has the synsets {software, software system, software package, package} and also nouns in WordNet are organised in a hierarchical tree structure based on hypernym and hyponymy. The hyponym of a noun is its subordinate, and the relation between a hyponym and its hypernym is an 'is a kind of' relation. The expanded terms obtained after finding all possible senses can be used for topic model generation and word construction based on sampling techniques using the LDA.

LDA is a generative probabilistic model for collections of discrete data such as text corpora (Blei et al., 2003). LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modelled as a finite mixture over an underlying set of topics. Each topic is, in turn, modelled as an infinite mixture over an underlying set of topic probabilities. In the context of text modelling, the topic probabilities provide an explicit representation of a document.

4.1 Finding the topics of corpus

Our objective is to find the topic of the corpus. For this, we find out the topic of the corpus by using Mallet implementing LDA (Momno, 2004) and then by Gensim implementing LSA. To get the topic of the corpus first we have to train the models in both Mallet and Gensim. Then we find out the topics of the corpus. In first test, the number of topics for the corpus is given as 10 then it is increased up to 350 topics for the corpus. Now the job is to find out the most popular words as there are many words in each of the topic that we have got as output. For example, there are 15 words that represent the topic 1. So it is required to find out the top words out of these words from all the topics. For this, we have done a frequency measurement of the words and the top words are selected from each test. These words truly represent the corpus. Then these top words from LDA and from LSA are compared using different similarity measures and the results that we got are:

Table 1 The comparison of topics between LDA and LSA (see online version for colours)

Topic 25		*Topic 50*		*Topic 100*	
Mallet	*LSA*	*Mallet*	*LSA*	*Mallet*	*LSA*
scientist	scientist	number	university	say	say
found	Found	scientist	scientist	human	university
system	journal	human	study	work	scientist
year	researcher	researcher	human	animal	human
researcher	Say	planet	journal	expert	found
earth	Study	material	found	researcher	journal
animal	One	risk	researcher	other	researcher
study	People	year	team	found	study
expert	university	result	one	scientist	cell
group	Other	team	year	way	way
Topic 200		*Topic 300*		*Topic 350*	
Mallet	*LSA*	*Mallet*	*LSA*	*Mallet*	*LSA*
say	Say	say	say	Say	say
researcher	Show	researcher	scientist	researcher	scientist
found	human	need	human	scientist	human
report	scientist	scientist	researcher	human	researcher
study	university	show	show	found	University
way	Year	found	part	expert	found
people	researcher	show	animal	show	expert
scientist	Study	human	study	study	Show
human	Cell	animal	year	way	study
show	found	study	found	University	Year

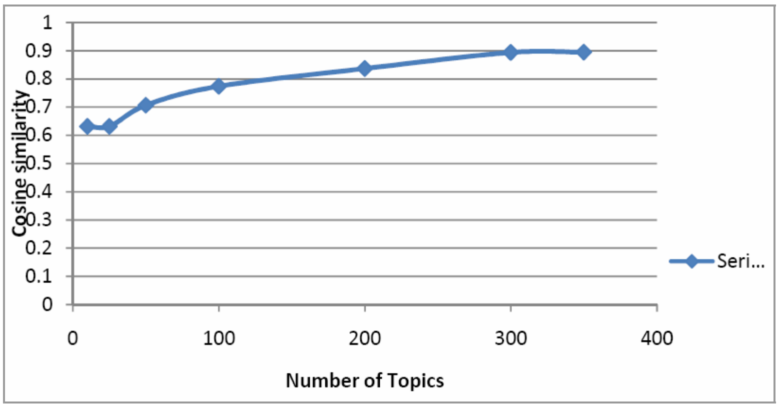
Note: The yellow marking shows the words that are common.

- Cosine similarity:

Table 2 Cosine similarity vs. number of topics for Mallet (LDA) and Gensim (LSA)

Cosine similarity plot	
Topic	*Cosine similarity*
10	0.632
25	0.632
50	0.707
100	0.774
200	0.837
300	0.894
350	0.895

Figure 1 No. of topics vs. cosine similarity (see online version for colours)

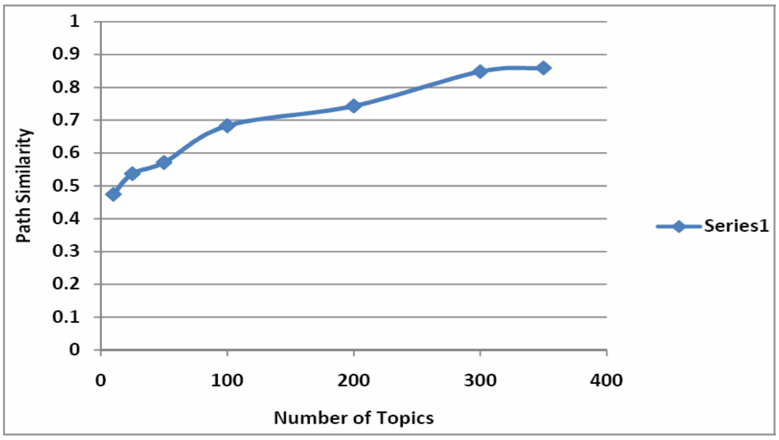


- Path similarity:

Table 3 Path similarity vs. number of topics for Mallet (LDA) and Gensim (LSA)

Path similarity plot	
Topic	*Path similarity*
10	0.474
25	0.537
50	0.571
100	0.683
200	0.743
300	0.848
350	0.859

Figure 2 No. of topics vs. path similarity (see online version for colours)

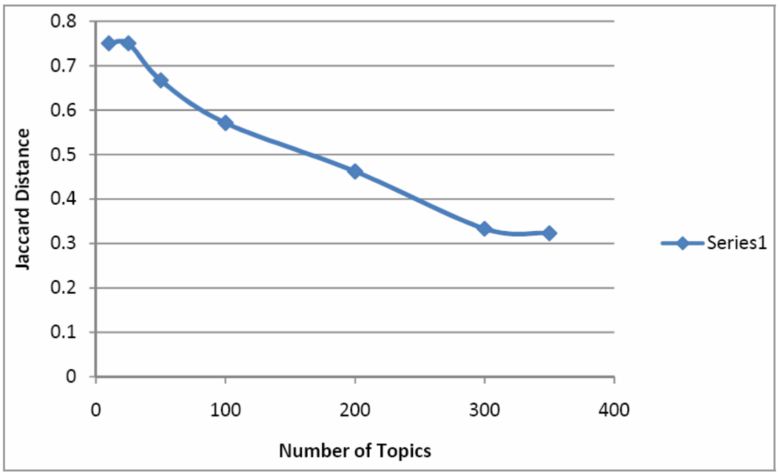


- Jaccard distance:

Table 4 Jaccard distance vs. number of topics for Mallet (LDA) and Gensim (LSA)

Jaccard distance plot	
Topic	*Jaccard distance*
10	0.75
25	0.75
50	0.667
100	0.571
200	0.462
300	0.333
350	0.323

Figure 3 No. of topics vs. Jaccard distance (see online version for colours)

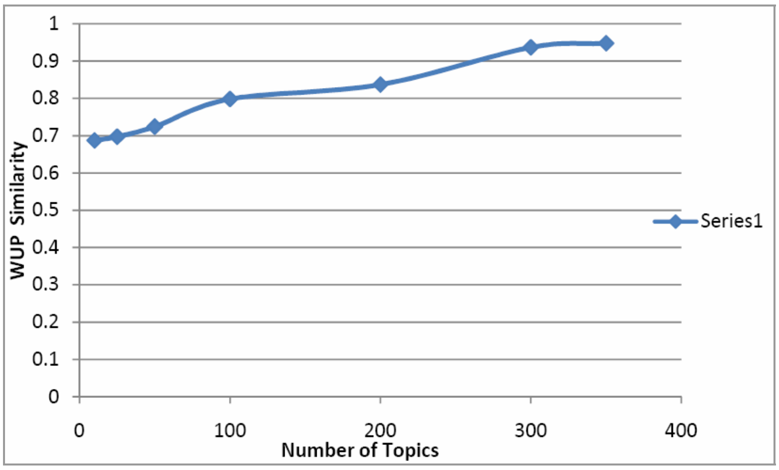


- WUP similarity:

Table 5 WUP similarity vs. number of topics for Mallet (LDA) and Gensim (LSA)

WUP similarity plot	
Topic	*Path similarity*
10	0.687
25	0.697
50	0.724
100	0.798
200	0.837
300	0.937
350	0.948

Figure 4 No. of topics vs. WUP similarity (see online version for colours)



It is observed that as the number of topics is increased the similarity between results from LDA and LSA increases.

4.2 *Dividing the corpus*

In the second experiment, we divided the corpus into two parts. The first part contains 75% of the corpus and the second part has 25% of the corpus. The topics of each of these were found out using the same procedure we used above. The only difference is that for both the parts of the corpus we used Mallet implementing LDA to get the topics. After getting the topics the most frequent words were found out for each number of topics (e.g., 10, 25, ……, 350). Then these words were compared using different similarity measures and the results that we have found is:

Table 6 The comparison of topics between 75% of corpus and 25% of corpus (see online version for colours)

Topic 10		*Topic 25*		*Topic 50*		*Topic 100*	
75% of corpus	*25% of corpus*	*75% of corpus*	*25% of corpus*	*75% of corpus*	*25% of corpus*	*75% of corpus*	*25% of corpus*
only	Researcher	say	say	say	say	say	say
being	Say	found	human	scientist	human	researcher	researcher
cell	Cell	scientist	other	human	show	human	other
say	Human	human	system	like	animal	other	new
human	Animal	researcher	researcher	researcher	scientist	animal	human
other	Group	other	group	show	researcher	two	group
year	Other	show	cell	help	help	new	scientist
new	Year	group	show	thing	group	scientist	part
study	People	expert	material	system	report	part	people
show	University	day	risk	way	like	work	work

Table 6 The comparison of topics between 75% of corpus and 25% of corpus (continued) (see online version for colours)

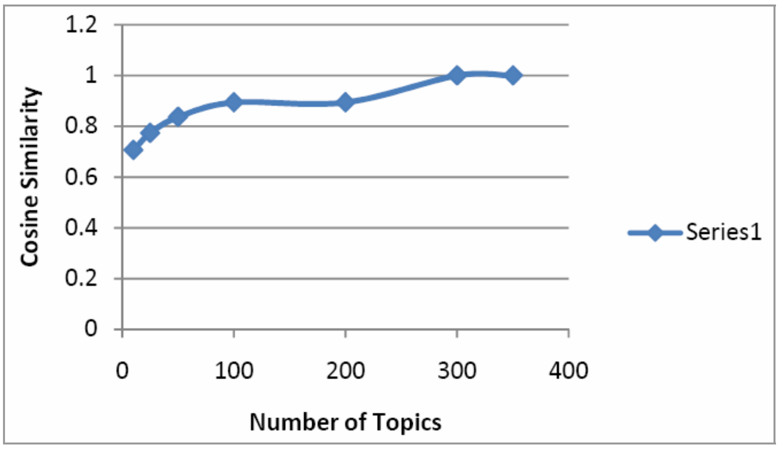
Topic 200		*Topic 300*		*Topic 350*	
75% of corpus	*25% of corpus*	*75% of corpus*	*25% of corpus*	*75% of corpus*	*25% of corpus*
say	Say	say	say	say	say
other	Other	researcher	other	researcher	other
found	After	show	one	show	researcher
researcher	Found	found	found	found	show
scientist	New	human	new	human	new
human	Scientist	new	researcher	new	one
first	researcher	other	show	other	found
work	Human	one	human	one	human
group	Work	group	group	group	group
need	Need	like	like	like	like

- Cosine similarity:

Table 7 Cosine similarity vs. number of topics for 75% and 25% of corpus

Cosine similarity plot	
Topic	*Cosine similarity*
10	0.707
25	0.774
50	0.837
100	0.894
200	0.894
300	1
350	1

Figure 5 No. of topics vs. cosine similarity (see online version for colours)

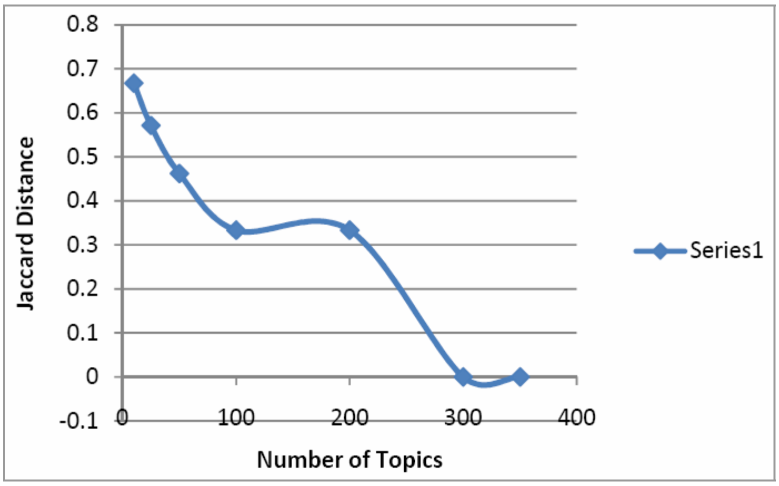


- Jaccard distance:

Table 8 Jaccard dist. vs. number of topics for 75% and 25% of corpus

Jaccard distance plot	
Topic	*Jaccard distance*
10	0.667
25	0.571
50	0.462
100	0.333
200	0.333
300	0
350	0

Figure 6 No. of topics vs. Jaccard distance (see online version for colours)

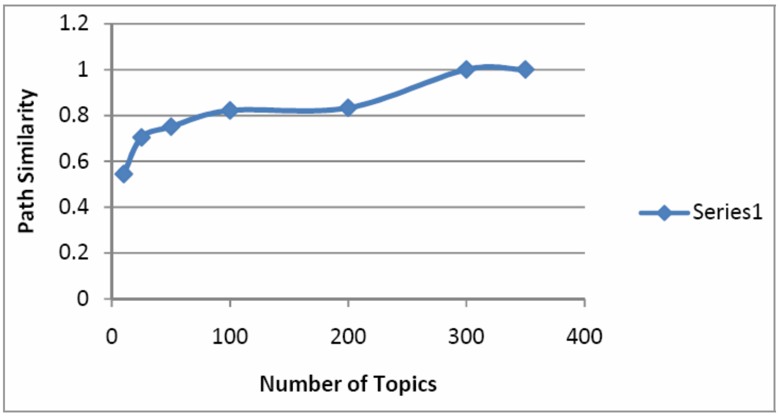


- Path similarity:

Table 9 Path similarity vs. number of topics for 75% and 25% of corpus

Path similarity plot	
Topic	*Path similarity*
10	0.545
25	0.704
50	0.751
100	0.822
200	0.833
300	1
350	1

Figure 7 No. of topics vs. path similarity (see online version for colours)

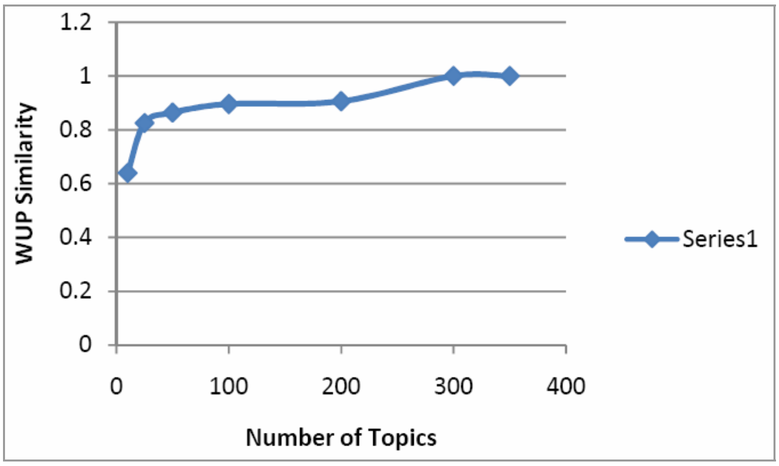


- WUP similarity:

Table 10 WUP similarity vs. number of topic s for 75% and 25% o f corpus

WUP similarity plot	
Topic	*WUP similarity*
10	0.64
25	0.825
50	0.864
100	0.896
200	0.906
300	1
350	1

Figure 8 No. of topics vs. WUP similarity (see online version for colours)

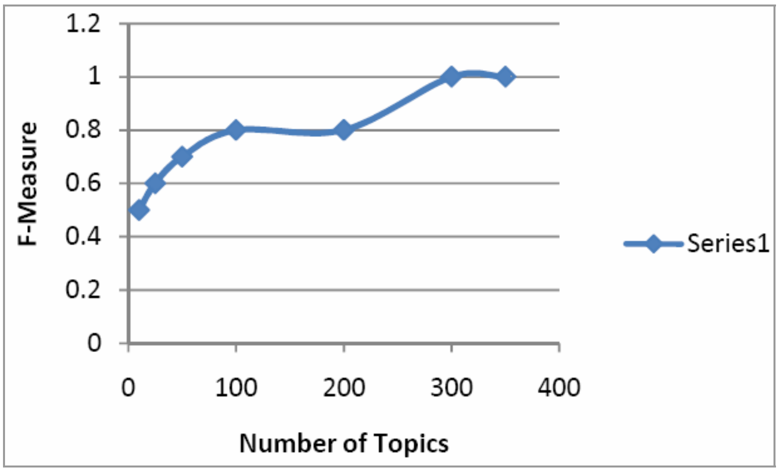


- F-measure:

Table 11 F-measure vs. number of topics for 75% an d 25% of corpus

F-measure plot	
Topic	*F-measure*
10	0.5
25	0.6
50	0.7
100	0.8
200	0.8
300	1
350	1

Figure 9 No. of topics vs. F-measure (see online version for colours)



It is evident from the first experiment that when we increase the number of topics we used to get better results. As the number of topics that are common from both LDA and LSA are much higher than when lesser number of topics is selected.

So whenever we intend to find out the topic of a corpus it is important to give more number of topics because then we can expect better results. It is also required that after getting the topics we do a word-frequency measurement to find out the words which are most common and truly represent the corpus. It would be better if we select 300 or above number of topics for a medium size corpus.

In the second experiment, we see that for a corpus related to same subject it is not required to select the entire corpus we can also get quite good result with selecting 75% or 80% of the corpus. For a very big corpus it will save time and resources. For this experiment also we get better results when the number of topics is around 300. So in our algorithm, we select number topics are 300 as it is optimum.

4.3 Proposed algorithm

In this paper, we propose an algorithm for document-document similarity by expansion of query for all the possible senses. We further use our proposed algorithm for computing the semantic distance measures to compute semantic similarity for identifying topic of unlabelled text from set of trained labelled data using topic modelling.

We name this algorithm as DDSSA. Pseudo code of the algorithm is as follows:

```
Input: two documents D1and D2
Output: Similarity measure based on semantic distance.
Algorithm:
STEP 1: Obtain the conceptual terms from documents D1($T_1$, $T_2$, $T_3$.........$T_n$) and
        D2($t_1$, $t_2$, $t_3$....$t_n$) using Topic Modelling for 300 topics.
STEP 2: Filter the conceptual terms by removing all possible prepositions, conjunctions,
        articles, special characters and other sentence delimiters.
STEP 3: Expand the conceptual terms into logically similar word (same sense) to form the
        synset (collection of similar words) as D1(S1,S2,…$S_{n1}$) and D2(s1,s2,…$s_{n2}$).
STEP 4: For all terms (Ti) in synsets of D1
            For all terms (Tj) in synsets of D2
                1  Find the most common part-of-speech for Ti based on its polysemy
                   count and return the part-of-speech for the version of Ti with the most
                   different senses.
                2  Find the minimum distance between any two senses for Ti and Tj in the
                   WordNet Tree based on the part-of-speech calculated in 1. This is a
                   normalized value between 0 and 1, returning 0 if the terms are exactly
                   the same and 1 if there is no similarity at all.
            End For
        End For
STEP 5: Construct the adjacency matrix for the relationship between the terms calculated in
        STEP 4 above. For instance, elements of Synset of D1 are (S1, S2, S3) and
        elements of Synset of D2 are (s1, s2). The Semantic Distances between D1 and D2
        are shown in the following matrix (‘dis’ is the function that computes the semantic
        distance).
        $$\begin{array}{cccc} & & \textit{Document 1} & \\ \textit{Document 2} & dis(S1,s1) & dis(S2,s2) & dis(S3,s1) \\ & dis(S1,s2) & dis(S2,s2) & dis(S3,s2) \end{array}$$
STEP 6: Compute the average distance for Synsets of D1with respect to synsets of D2.The
        distance between D1,D2 using DDSSA is calculated as,
        $$avgDist = \frac{\min\{dis(S1,s1), dis(S2,s1), dis(S3,s1)\} + \min\{dis(S1,s2), dis(S2,s2), dis(S3,s2)\}}{Number\ of\ synsets\ for\ Document\ 2\ (2\ in\ this\ case)}$$
STEP 7: If avgDist is zero
            Return Documents are same.
        Else If avgDist is less than a predefined threshold
            Return Documents are nearly same.
        Else
            **Return** Documents are different.
        *End If*
```

Explanation

For instance, let us consider a document based on planet Uranus which has synsets namely Uranus, planet, solar, moons, cloud and another document which is based on planet Neptune which has synsets namely Neptune, planet, sun. We construct the adjacency matrix based on the semantic distance. The semantic distance is computed using taxonomy-based shortest path categorisation based on sine similarity.

4.4 Use of DDSSA in topic modelling

Classifying text is a problem of great interest, and has received much attention in recent years. This is due to a number of coinciding factors:

a *volume* – the volume of digitised text available is growing seemingly exponentially because of the internet's explosive expansion and business steadily progressing transition from paper documents to electronic ones

b *computer speed* – text classification requires a fair amount of horsepower, especially when a large number of documents are to be analysed or classified

c *need* – text classification is of great practical value; as the sheer volume of electronic documents we interact with each day increases, we are approaching the point where it is no longer feasible for us to deal with them manually.

Many machine learning algorithms have been applied to the text-classification by supervised learning (Ko and Seo, 2009). The supervised learning algorithm finds a representation or decision rule from an example set of labelled documents for each class. A wide range of the supervised learning algorithms has been applied to this area using a training dataset of labelled documents.

Common among many of these research efforts is an approach to building hierarchical text classifiers by first subdividing the task into a number of smaller classification tasks – one per decision point in the hierarchy – then building a separate classifier for each of the smaller tasks (Boyapati, 2002).

An unlabeled document does not contain most important piece of information (i.e., its class). Here is an intuitive example where unlabeled data might be useful. Suppose we are interested in recognising web pages about academic courses. We have with us a few known classes and non-classes web pages, along with a large number of web pages that are unlabeled. In labelled data, we find that pages containing the word 'homework' tend to be about academic courses. Suppose we use this fact to estimate classification of an unlabelled examples, we observe that the word 'classnotes' which occurs frequently in the given unlabeled examples that are now believed to belong to the positive class. This co-occurrence of the words like 'homework' and 'class notes' over the large set of unlabeled training data can provide useful information to construct a more accurate classier that considers both homework and class notes as indicators of positive examples. In this paper, we discuss about using of trained data obtained from labelled data can be used to classify an unlabelled data. We have a labelled data on planet Uranus which we use as training data for the machine learning using sampling techniques based on LDA. We obtain the concepts, concept score for the given training data which ranges between 0 to 400. We obtain the concepts for concepts score greater than 300 by ranking the concepts in descending based on concept scores from given labelled text. Store the output as trained dataset to be used later to classify the unlabelled text.

Table 12 Concepts, concept score obtained from topic 'Uranus'

Topic	*Concepts, concept score*
Uranus (Planet)	Uranian, 388
	Rings, 388
	Solar, 387
	Planet, 382
	System, 371
	Exosphere, 371
	Hydrocarbons, 360
	Orbit, 337

The corpus-level parameters α and β are assumed to be sampled once in the process of generating a corpus. In our experiment, the value of $\alpha = 0.125$ and $\beta = 0.1$ are used for corpus generation. We also use other labelled text and result can be stored in the form of dictionary for faster access and effective retrieval. This reduces the time required to retrieve the topic and corresponding concepts associated with it. Hence, it reduces time complexity and improves the performance of overall process. Suppose we have unlabelled text for which the class is not known. We apply the same machine learning techniques to obtain concept and concept score. After ranking of the concepts based on the concept score in descending order, we extract those concepts whose concept score are greater than 300. We assume that concepts for which concept score are greater than 300 are most nearly related to given labelled or unlabelled text.

We consider the semantic similarity measures to compute the similarity between concepts for which we use our proposed algorithm as mentioned above named DDSSA. We compute the semantic distance between the concepts obtained from unlabelled data with the concepts obtained different trained dataset. The greater the value of semantic distance more is the semantic similarity between the two concepts. Hence, this approach can be extended to use of the term-based expansion for attaining large datasets and hence provide effective topic identification of the given unlabelled data.

Table 13 Concepts, concept score obtained from topic 'Hill'

Topic	*Concepts, concept score*
Hill (Australian Cricketer)	Tournament, 392
	Cricket, 390
	Record, 388
	Cup, 341
	Run, 336
	Centuries, 321
	Team, 309

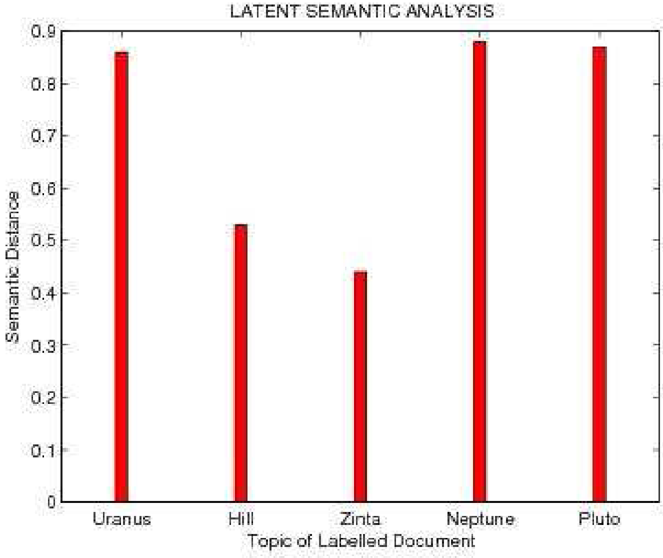
5 Experimental results and discussions

We applied the above algorithm for 15,000 unlabelled documents taken from NLTK and MALLET datasets, where each of the documents contains nearly average of 5,000 terms respectively. We had conducted simulation in order to evaluate our DDSSA. After several simulation runs, it was found that for documents which were relatively similar to each other had their semantic distance between the ranges of 0.6 to 1. Few results of our approach are tabulated in Table 14.

Table 14 Comparison of proposed algorithm without WUM and with WUM to LSA

Unlabelled document	*Labelled document*	*Latent semantic analysis*	*Proposed algorithm without web usage mining*		*Proposed algorithm with web usage mining*	
		Without term expansion	*With term expansion*	*Without term expansion*	*With term expansion*	*Without term expansion*
	Uranus (Planet)	0.86	0.7733	0.6531	0.7997	0.8241
	Hill (Australian Cricketer)	0.53	0.313	0.3485	0.425	0.3969
	Zinta (Indian Actress)	0.44	0.3092	0.398	0.4396	0.4169
	Neptune (Planet)	0.88	0.4485	0.47	0.403	0.4551
	Pluto (Planet)	0.87	0.3393	0.4028	0.4762	0.4727

Figure 10 Semantic distance obtained using LSA (see online version for colours)



As we know that LSA fails to handle the words with multiple meaning effectively and requires high amount of computational time as it depends on SVD. Each time when a document is updated, matrix need to be recomputed and hence it increases the time complexity. We overcome some of the limitations of LSA by using hypernym and synonym to compute all possible meaning of a given word for obtaining better concepts which best represent the document.

Figure 11 Semantic distance obtained using proposed algorithm without web usage mining and without term expansion (see online version for colours)

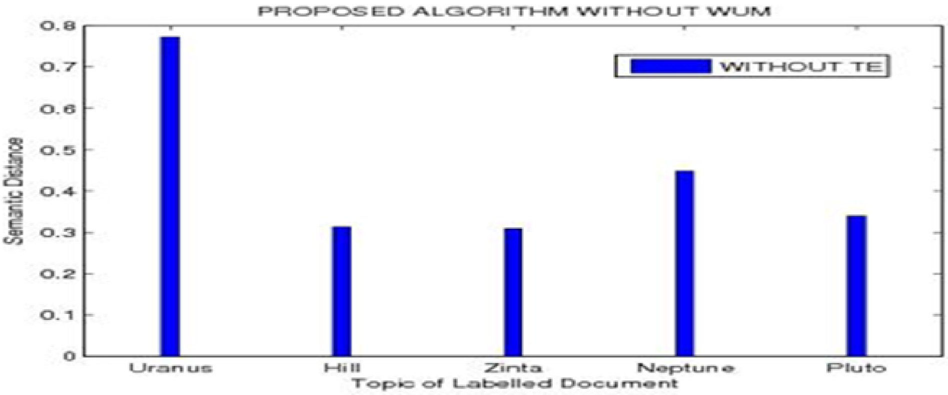
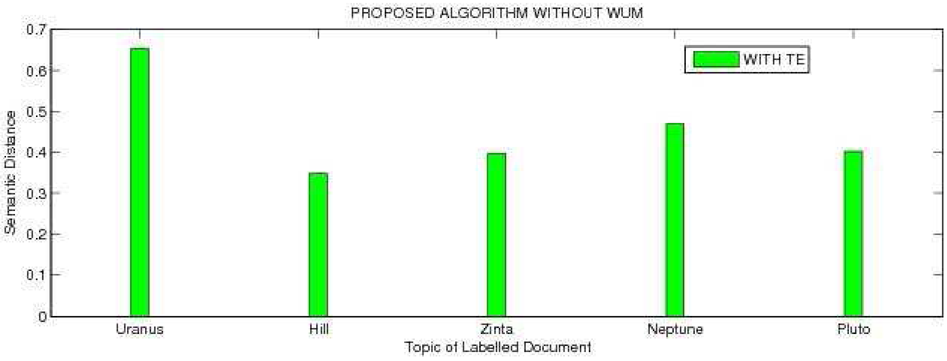
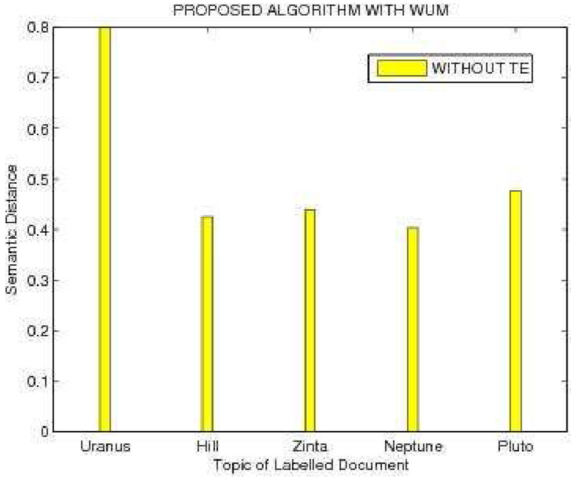


Figure 12 Semantic distance obtained using proposed algorithm without web usage mining and with term expansion (see online version for colours)



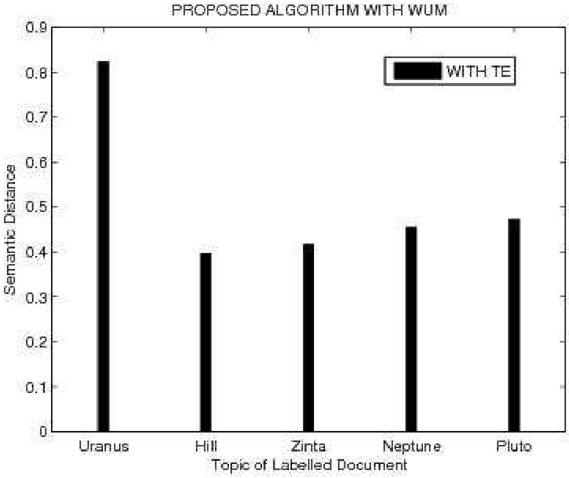
We observe that the unlabelled document has greater semantic similarity with respect to labelled document Uranus. Moreover LSA fails to distinguish the nearly similar as well as dissimilar documents. As we input the unlabelled document which is much related to Uranus, we find from the given table that LSA shows similarity between the topic of unlabelled document on Uranus and labelled document on Neptune and Pluto which are not the actual similar.

Figure 13 Semantic distance obtained using proposed algorithm with web usage mining and without term expansion (see online version for colours)



Our proposed approach overcomes the drawback by identifying the similar, nearly similar documents as well as distinguishing between dissimilar documents in a much better manner and also the time complexity is reduced as we use the conceptual terms and concept weight for computing the document similarity. Hence, it can better identify the class of unlabelled document to which it belongs to. So from the above table we conclude that topic of unlabelled document is much closely related to the topic of labelled document (i.e.) Uranus. It is also found that when the proposed algorithm when integrated with web usage mining can fetch better results.

Figure 14 Semantic distance obtained using proposed algorithm with web usage mining and with term expansion



It is observed that semantic distance increases after term expansion as we obtain all the possible sense of the conceptual terms which may or may not be relevant to given text.

Since unlabelled text may be short text, hence it would be useful in short text classification. The performance of our approach can be enhanced effectively by extracting part of the document from the given user query.

6 Conclusions and future work

The performance of our approach can be improved when we have large synset for computing semantic measures and dictionary words are sufficiently large. Thus, web log analysis when integrated with text classification can give better and selective results and reduce the overall time complexity of the retrieval process and hence improve the performance of the entire process. Our proposed algorithm can be used in the process of query expansion to compute query-query similarity and grouping the similar queries into a cluster based on user log based analysis. Hence, it can be used by search engine for generating nearly accurate results.

References

April, K. and Pottenger, W.M. (2006) 'A framework for under-standing latent semantic indexing performance', *J. Inf. Process. Manag.*, Vol. 42, No. 1, pp.56–73.

Bhogal, J., Macfarlane, A. and Smith, P. (2007) 'A review of ontology based query expansion', *Information Processing and Management, Science Direct*, Vol. 43, No. 4, pp.866–886.

Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003) 'Latent Dirichlet allocation', *Journal of Machine Learning Research*, Vol. 3, pp.993–1022.

Boyapati, V. (2002) 'Improving hierarchical text classification using unlabelled data', *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, IGIR'02*, 11–15 August.

Chan, P.K. (1999) 'A non-invasive learning approach to building web user profiles', *KDD-99 Workshop on Web Usage Analysis and User Profiling.*

Fellbaum, C. (Ed.) (1998) *WordNet, 'An Electronic Lexical Database'*, MIT Press, Cambridge, USA.

Fonseca, B.M., Golgher, P., Pôssas, B., Ribeiro-Neto, B. and Ziviani, N. (2005) 'Concept-based interactive query expansion', *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, ACM.

Gong, Z., Chan, W.C. and Leong, H.U. (2005) 'Web query expansion by WordNet', *DEXA, LNCS*, Vol. 3588, p.166.

Imran, H. and Sharan, A. (2009) 'Thesaurus and query expansion', *International Journal of Computer Science & Information Technology*, Vol. 1, No. 2, pp.89–97.

Ko, Y.j. and Seo, J.y. (2009) 'Text classification from unlabeled documents with bootstrapping and feature projection techniques', *Information Processing and Management, Science Direct*, Vol. 45, No. 1, pp.70–83.

Koller, D. and Sahami, M. (1998) 'Hierarchically classifying documents using very few words', *Proceedings of the14th international Conference on Machine Learning ECML98.*

Metzler, D., Dumais, S. and Meek, C. (2007) 'Similarity measures for short segments of text', *Proceeding ECIR'07 Proceedings of the 29th European Conference on IR Research*, Springer-Verlag Berlin, Heidelberg.

Mining, C.D., Raghavan, P. and Schütze, H. (2009) *An Introduction to Information Retrieval*, Cambridge University Press, Cambridge, England.

Mladenic, D. (1998) *Machine Learning on Non-Homogeneous, Distributed Text Data*, PhD thesis, University of Ljubljana, Slovenia.

Momno, D. (2004) *Machine Learning with MALLET*, Information on Extraction and Synthesis Laboratory, Department of CSU Mass, Amherst.

Noah, S.A., Amruddin, A.Y. and Omar, N. (2007) 'Semantic similarity measures for Malay sentences', *ICADL 2007, LNCS*, Vol. 4822, pp.117–126.

Pantelm, P. and Lin, D. (2002) 'Discovering word senses from text', *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Parwardhan, S., Banerjee, S. and Pedersen, T. (2003) 'Using measures of semantic relatedness for word sense disambiguation', in *Proceeding of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*.

Qiu, Y.g. and Frei, H.P. (1993) 'Concept based query expansion', *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM.

Sahami, M. and Heilman, T. (2006) 'A web-based kernel function for measuring the similarity of short text snippets', in *Pro. of WWW 2006*, pp.377–386.

Scott, S. and Matwin, S. (1998) 'Text classification using WordNet hypernyms', *Coting-ACL'98 Workshop: Usage of WordNet in Natural Language Processing Systems*, August, pp.45–51.

Sebastiani, F. (2002) 'Machine learning in automated text categorization', *ACM Comput. Survey*, Vol. 34, No. 1, pp.1–47.

Wetzker, R., Alpcan, T., Bauckhage, C., Umbrath, W. and Albayrak, S. (2007) 'An unsupervised hierarchical approach to document categorization', *Proceeding WI '07 Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*.

Xhemali, D., Hinde, C.J. and Stone, R.G. (2009) 'Naïve Bayes vs. decision trees vs. neural network classification of training web pages', *International Journal of Computer Science Issues*, Vol. 4, No. 1, pp.16–23.