SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
BENGALURU 562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Technical Seminar Entitled

"Semantic Documents Relatedness using Concept Graph Representation"

Submitted by

KARAN SAXENA	**1MV13CS047**
AMIT ASISH BHADRA	**1MV13CS014**

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgavi – 590018, Karnataka State, India

Technical Seminar Entitled

"Semantic Documents Relatedness using Concept Graph Representation"

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

For the academic year 2016-2017

Submitted by:

Karan Saxena **1MV13CS047**

Amit Asish Bhadra **1MV13CS014**

Carried out at

Sir M. Visvesvaraya Institute of Technology

Bangalore - 562157

Under Guidance of

Prof. Dilip K. Sen

Professor and Head, Dept. of Computer Science

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BENGALURU – 562157

SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belgavi)

Bangalore – 562157

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the technical report entitled **"Semantic Documents Relatedness using Concept Graph Representation"** is a bonafide work carried by **Karan Saxena (1MV13CS047)** and **Amit Asish Bhadra (1MV13CS014)** in partial fulfillment for the degree of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belgaum** during the **academic year 2016 – 2017.** The report has been hereby approved as it satisfies the academic requirements in respect of the project work prescribed for the course of Bachelor of Engineering Degree.

……	……
Signature of the guide	**Signature of the HOD**
Prof. Dilip K. Sen	**Prof. Dilip K. Sen**
Prof. and Head, Dept. of CSE	**Prof. and Head, Dept. of CSE**

Examiners:

1. ……

2. ……

ACKNOWLEDGEMENT

The successful completion of any work depends upon the cooperation and help of many people and not just those who directly execute the work. It is difficult to express in words our profound sense of gratitude to those who helped us, but we make a humble attempt to do so.

I express my sincere gratitude to our **Principal Dr. M.S.Indira**, Sir MVIT for providing facilities.

I wish to place on record my sincere thanks to **Prof Dilip K. Sen, Head of the Department**, Computer Science and Engineering for encouragement and support, and for giving us the valuable suggestions regarding the project.

We also thank the members of the faculty of CSE department, whose suggestions helped us in the course of this project.

Our heartfelt thanks to all the above mentioned people who have contributed in the accomplishment of this project.

KARAN SAXENA (1MV13CS047)

AMIT ASISH BHADRA (1MV13CS014)

ABSTRACT

Semantic relatedness, or similarity between documents plays an important role in many textual applications such as information retrieval, document classification and clustering, question answering and more. Measurement of semantic relatedness comprises two constituents: an effective representation of documents, and a similarity measure between documents in terms of their respective representations.

We explore this issue of document similarity by experimenting with various existing language models, examining their performance in the task of computing document similarity. In particular, we chose four semantic models (LSA, LDA, Word2Vec and Doc2Vec) and one frequency-based model (TfIdf), for extracting document features.

We deal with the problem of document representation for the task of measuring semantic relatedness between documents. The growing popularity of Semantic Web, Wikipedia, DBpedia and other universal knowledge bases, brought a new type of representing vectors, where the i^{th} component reflects the relative weight, or relevance of the i^{th} concept of the knowledge base in the represented document. Thus, the length of these concept vectors equals the number of concepts in the knowledge base, which could be in the millions. Nevertheless, the simple correspondence between components and concepts makes these vectors intuitive, easier for human interpretation.

A document is represented as a compact concept graph where nodes represent concepts extracted from the document through references to entities in a knowledge base such as DBpedia. Edges represent the semantic and structural relationships among the concepts. Several methods are presented to measure the strength of those relationships. Concepts are weighted through the concept graph using closeness centrality measure which reflects their relevance to the aspects of the document. A novel similarity measure between two concept graphs is presented. The similarity measure first represents concepts as continuous vectors by means of neural networks. Second, the continuous vectors are used to accumulate pairwise similarity between pairs of concepts while considering their assigned weights. This method outperforms state-of-the-art methods, while the concept graphs are much smaller than the concept vectors generated by other models.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Most document representations map the documents to vectors of a fixed length, aiming to map semantically related documents to close vectors in the vector space. The simplest representation of these is the bag-of-words, where a document is represented as a vector of frequencies organized with respect to a vocabulary of words. Component i of the vector reflects the frequency, in the represented document, of the i^{th} word of the vocabulary. Bag-of-words only represents the syntax of the words and their frequencies. It does not address multiple meanings of same word or synonymy of words. It also totally ignores the order of the words in the document.

Latent topic models, such as Latent Semantic Analysis, Latent Dirichlet Allocation, or Word2Vec, map words and documents to vectors whose components represent latent topics. The (configurable) length of those vectors is much smaller than with bag-of-words. The latent topics, however, are hard to interpret, they are not as intuitive as the components of the bag-of-words vectors.

The growing popularity of Semantic Web, Wikipedia, DBpedia and other universal knowledge bases, brought a new type of representing vectors, where the i^{th} component reflects the relative weight, or relevance of the i^{th} concept of the knowledge base in the represented document. Thus, the length of these concept vectors equals the number of concepts in the knowledge base, which could be in the millions. Nevertheless, the simple correspondence between components and concepts makes these vectors intuitive, easier for human interpretation.

The representations and similarity measurements described above have some shortcomings. The bag-of-words model lacks reflection of the semantics of the text. The topic models (both latent and explicit) do not consider the structure and relationships among the topics or concepts used for the representation.

We discuss a novel document representation and a measurement of similarity that combine the advantages of them all. Specifically, this method:

i) Uses topic models that are based on explicit concepts

ii) Considers the relationships among the concepts and iii) uses neural network based methods to represent concepts (as opposed to words) as continuous vectors.

For document representation we build a graph whose nodes are explicit concepts from a knowledge base. We extract the concepts directly from the document text using a mention detection tool such as SpotLight or TagME. We observe that some concepts may be closely related to the main aspects of the document while others may drift. To capture the relative importance, or weight, of the detected concepts, we arrange them as nodes in a concept graph, and use the knowledge base to assign weights to edges between each pair of concepts. We consider different types of relationships and structural links among the concepts and their environment in the knowledge base. We argue that concepts with higher coherence are more important to the aspects of the document. To evaluate the coherence of each concept with other concepts, we use the concept's level of centrality, which measures its closeness, through the weighted edges of the concept graph, to the rest of the concepts. We take the concept's centrality level to be its weight in the concept graph.

To evaluate semantic relatedness between two documents, we measure the similarity between their concept graphs as the weighted sum of contributions from all pairs of nodes, one from each graph, of their pairwise similarity. The weights are the weights of the nodes in their respective concept graphs. We discuss several pairwise similarity measures, one of which is a novel measure, for which we propose Concept2Vec that expands upon Word2Vec to represent a concept as a continuous vector of 200 dimensions.

1.2 History

In order to algorithmically perform a comparison of two documents, it is necessary to transform the original documents (plain text) into a representation, which a computer can understand, i.e. a vector of features. The main problem is, what type of features to use. It is worth noting, that a common step for all models mentioned here is a preprocessing step, where the input string is tokenized into words and each word is lemmatized. The result is a set of terms corresponding to the input document. These terms can be used in several ways for constructing a model of the document. Among some of the more basic models are:

- Boolean model - equals to 1 if a term t occurs in document d and 0 otherwise
- Term frequency tf (t, d) - raw number of times that term t is in the document
- Logarithmically scaled term frequency log(tf (t, d)+ 1)
- Augmented frequency - to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document

There is also the possibility of utilizing external linguistic resources, such as WordNet, for processing synonyms or other semantic information. However, we want to minimize the dependency of the methods on any language-specific tool.

If not specified otherwise the similarity score for each pair of documents is computed as the cosine similarity between two feature vectors v1 and v2:

$$sim(v_1, v_2) = \frac{\sum_{i=1}^{m} v_1[i] * v_2[i]}{\sqrt{\sum_{i=1}^{n} v_1[i]^2} * \sqrt{\sum_{i=1}^{n} v_2[i]^2}},$$

CHAPTER 2

TECHNICAL DESCRIPTION

2.1 Tf-Idf

The TfIdf (Term frequency - Inverse document frequency) weight of a term is a statistical measure used to evaluate how important a word is to a document in a collection or a corpus. Its importance increases proportionally to the number of times the term appears in the document, but is offset by its frequency in the corpus. The TfIdf weight of a term is a product of its frequency tf(t, s) and inverse document frequency:

$$idf(t, D) = \log \frac{N}{|d \in D : t \in d|},$$

where D is the document set, N is the size of set D and |d ∈ D : t ∈ s| is the number of documents from D where the term t appears. The final value is then computed as:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D).$$

2.2 LSA

Latent Semantic Analysis (LSA), also known as Latent Semantic Indexing (LSI), is a method for extracting and representing the contextual meaning of words by statistical computations performed on a corpus of documents. The underlying idea is that the totality of information about all the word contexts, in which a given word does and does not appear, provides a set of mutual constraints that largely determines the similarity of meaning of words. The adequacy of LSA's reflection of human knowledge has been established in a variety of ways.

The creation of an LSA model starts with building a m×n matrix A, where n is the number of documents in the corpus and m is the total number of terms that appear in all documents. Each column of A represents a document d and each row represents term t.

There are several methods on how to compute the elements a_{td} of matrix A representing term frequencies, and among the most common are: Term frequency, TfIdf or Entropy.

With the matrix A built, LSA applies Singular Value Decomposition (SVD), which is defined as $A = U\Sigma V^T$, where $U = [u_{ij}]$ is an $m \times n$ matrix and its column vectors are called left singular vectors. Σ is a square diagonal $n \times n$ matrix and contains the singular values. $V^T = [v_{ij}]$ is an $n \times n$ matrix and its columns are called right singular vectors. This decomposition provides latent semantic structure of the input documents, which means, that it provides a decomposition of documents into n linearly independent vectors, which represent the main topics of the documents. If a specific combination of terms is often present within the document set, it is represented by one of the singular vectors.

2.3 LDA

Latent Dirichlet Allocation (LDA) can be basically viewed as a model which breaks down the collection of documents into topics by representing the document as a mixture of topics with their probability distributions. The topics are represented as a mixture of words with a probability representing the importance of the word for each topic.

Since LDA provides probability distributions of topics, it is possible to use statistical measures for quantifying the similarity between two documents. There are many such measures, like KL-divergence, Hellinger distance or Wasserstein metric to name just a few. We generally choose to use the Hellinger distance along with the cosine similarity to get optimum results.

2.4 Word2Vec

Word2Vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text. It comes in two flavors, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. Algorithmically, these models are similar, except that CBOW predicts target words (e.g. 'mat') from source context words ('the cat sits on the'), while the skip-gram does the inverse and predicts source context-words from the target words. This inversion might seem like an arbitrary choice, but statistically it has the effect that CBOW smoothes over a lot of the distributional information (by treating an entire context as one observation). For the most part, this turns out to be a useful thing for smaller datasets. However, skip-gram treats each context-target pair as a new observation, and this tends to do better when we have larger datasets.

Briefly, Word2Vec is a two-layer neural net published by Google in 2013. It implements continuous bag-of-words and skip-gram architectures for computing vector representations of

words, including their context. The skip-gram representation popularized by Mikolov has proven to be more accurate than other models due to the more generalizable contexts generated. The output of Word2Vec is a vocabulary of words, which appear in the original document, along with their vector representations in an n-dimensional vector space. Related words and/or groups of words appear next to each other in this space.

Since Word2Vec provides vector representations only for words, we need to combine them in some way to get a representation of the whole document. This can be done by averaging all the word vectors for the given document, and thus creating just one document vector, which can be compared to another by cosine similarity (model designated as 'Word2Vec'). It can be extended to an n-gram analogy (denoted as 'W2V-pn'), i.e. combining word vectors for phrases with n words and then computing similarities between these phrase-vectors from both input documents.

2.5 Doc2Vec

Google's Word2Vec project has created lots of interests in the text mining community. It provides high quality word vectors, however there is still no clear way to combine them into a high quality document vector. Doc2Vec (Paragraph2Vec) modifies the Word2Vec model into unsupervised learning of continuous representations for larger blocks of text, such as sentences, paragraphs or entire documents. An algorithm called Paragraph Vector is used on the IMDB dataset to produce some of the most state-of-the-art results to date. In part, it performs better than other approaches, because vector averaging or clustering lose the order of words, whereas Paragraph Vectors preserve this information. There is also an n-gram analogy, i.e. computing Paragraph Vectors for phrases of length n and then comparing those phrases from both input document. The original model, which computes vectors for the whole documents is denoted as 'Doc2Vec' and the n-gram analogies are denoted as 'D2V-pn'.

2.6 Concept2Vec

With the proliferation of Semantic Web, Wikipedia, DBpedia and other universal knowledge bases, which are continually expanded by the general crowd, and abundantly referenced as a source of high dimensional space of fine grained natural concepts, a new form of vector representation evolved, which reflects the knowledge base concepts that are

relevant to the represented document. The semantic meanings of a document, as expressed through these concepts, is easier for human interpretation.

The key idea of document representation by a concept graph is to link the concepts in the document text to entities of a knowledge base, and to explore the structural and semantic information among them in the knowledge base, to create a graph for these entities. Based on the observation that the core aspects of a document should be a set of closely related concepts, we measure the centrality, over the graph, of each concept to obtain its weight.

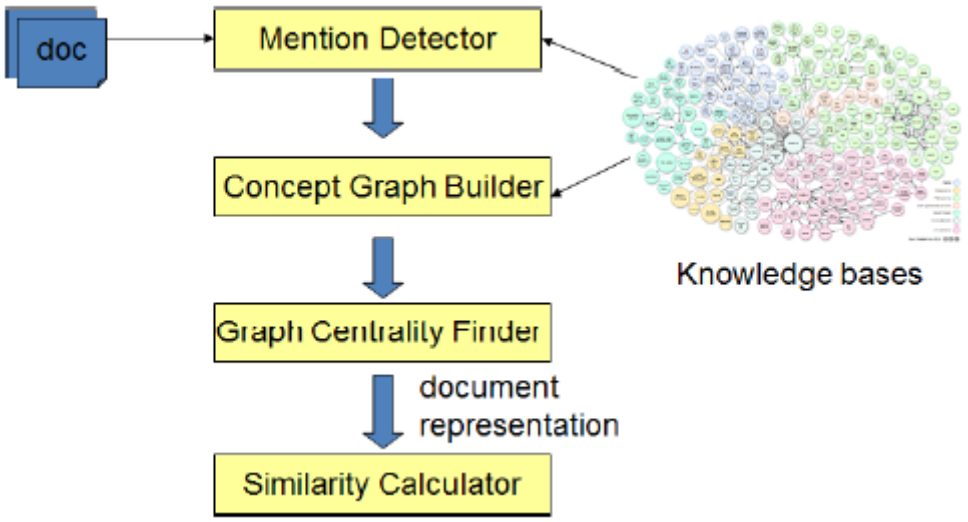

Fig 1: Architecture of the method

First, given the input document, we use an existing mention detection tool such as Spotlight or TagMe to detect the mentions in the document text. A mention is a term/phrase in the text that corresponds to a defined concept in the knowledge base. We use DBpedia as our knowledge base. The English version of DBpedia 2014 comprises 4.58 million concepts, each corresponds to a Wikipedia article. These are annotated with 583 million facts.

Given the following document text, for example, the tool will detect four mentions and their corresponding concepts (shown in brackets). Gambling [http://dbpedia.org/resource/Gambling] increases aggregate demand [http://dbpedia.org/resource/Aggregate_demand] for goods and services [http://dbpedia.org/resource/Goods_and_Services] in the economy [http://dbpedia.org/resource/Economy].

The nodes of the concept graph are the detected concepts in the given document. We measure the strength of association between each pair of concepts, based, as follows, on their features and their relationship in the knowledge base. Any pair of concepts with a non-zero association between them is connected by an edge in the concept graph, whose weight is

taken to be the strength of that association. The inter-concept association is calculated as a combination of three types of associations, each reflecting the extent of similarity, or relatedness of the concepts in the scope of one feature:

i) Context association: relative frequency of occurrence of both concepts in same contexts.

ii) Category association: how close the concepts are in the knowledge base taxonomy of categories

iii) Structure association: the related structural information from the knowledge base

Once the edges are determined, the centrality of each node in the concept graph is computed, and set to be the weight of the node.

Finally we define several measures of similarity between two concept graphs which aim to reflect the semantic relatedness of the documents they represent.

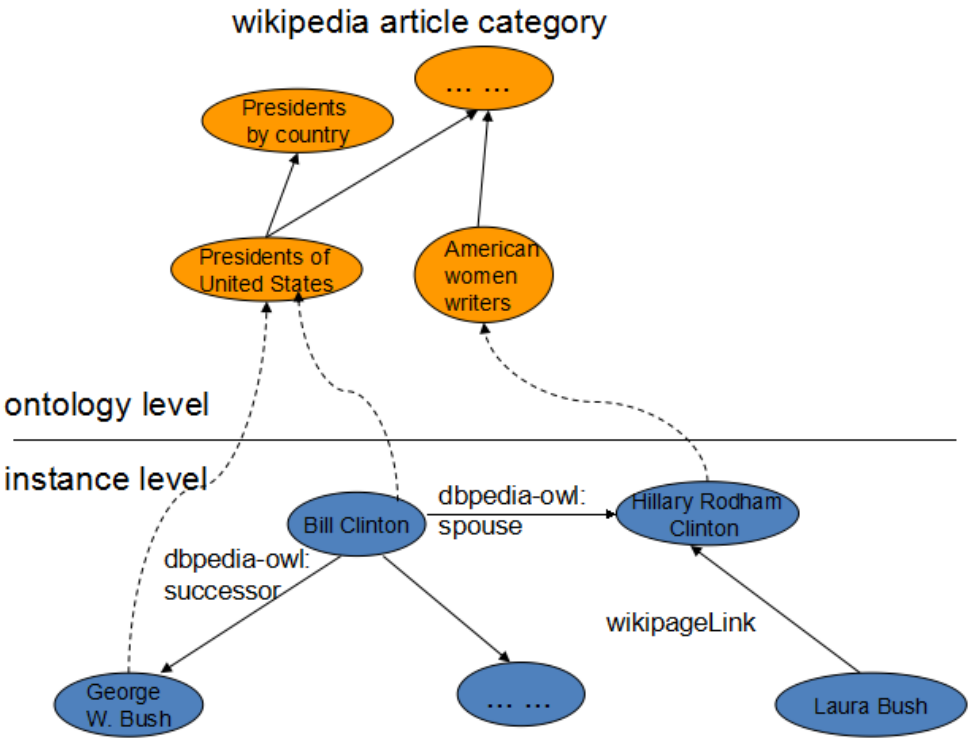

Fig 2: Reference DBpedia Data

2.6.1 Concept Graph Builder

Given a document, we first extract its concepts by employing a mention detection tool. The extracted concepts form the nodes of the concept graph. The edges of the graph represent the relationships between the concepts as derived from DBPedia, through the three kinds of association between concepts:

i) context association
ii) category association
iii) structure association.

In the following subsections, we describe how we measure the extent of these associations.

2.6.1.1 Context Association

Context association between two concepts reflects how often both concepts share contexts. In Wikipedia, we take shared contexts to be shared incoming links, i.e., pages that point to both concepts. A large number of shared incoming links indicates a high context association. Given two concept m1 and m2 we use the metric from to measure their context associations.

Let M1 and M2 denote the respective sets of incoming links to m1 and m2, then

$$ctxt(m_1, m_2) = 1 - \frac{\log(\max(|M_1|, |M_2|)) - \log(|M_1 \cap M_2|)}{\log|W| - \log(\min(|M_1|, |M_2|))}$$

where |W| is the total number of concepts in the knowledge base.

2.6.1.2 Category Association

DBpedia provides with category information for each concept, and a taxonomy of categories. Categories are intended to group together pages on similar subjects. Hence, if two concepts belong to the same Wikipedia category, they are associated with the same topic. Through the Wikipedia taxonomy of categories, we measure similarity of categories to evaluate the level of category association between concepts. Given two concepts m1 and m2, let C1 = {c11, c12, : : : , c1p}, and C2 = {c21, c22, : : : , c2q} denote the respective sets of categories that m1 and m2 belong to. We measure the category association between these two concepts by first finding the pairwise similarity between any two individual categories c1i and c2j, and then combining the pairwise category similarities between all pairs of categories one from C1 and the other from C2, to form the groupwise similarity between C1 and C2,

which is set to be the level of category association between the given two concepts. Next we define the workable pairwise similarity measures.

Pairwise category similarity: We calculate the similarity of two categories in a taxonomy based on the specificity, or the information content of the categories. First, the information content score IC(c) is computed for each node c in the taxonomy. Then given two nodes ci and cj in the taxonomy, let MSCA(ci; cj), denote the common ancestor of ci and cj with highest information content, then the pairwise similarity of ci and cj is computed as follows.

$$pairwise(c_i, c_j) = \frac{IC(MSCA(c_i, c_j))}{IC(c_i) + IC(c_j)}$$

2.6.1.3 Structure Association

Wikipedia infoboxes contain information about concepts and their various types of relationships, thus inducing a structural graph, G(V;E), over the concepts, whose edges e are labeled by predicates pred(e) that indicate the type of the relationship. We define the weight of an edge e to be

$$w(e) = \log(|pred(e)|) / \log(|E|)$$

where |pred(e)| is the frequency of pred(e) in E. The intuition is that frequent predicates represent a general, less significant relationship. We thus define the structure association between two concepts m1, and m2, as the inverse of the sum of the weights of the edges e1, e2, : : : , ek of the shortest path from m1 to m2:

$$struct(m_1, m_2) = \frac{1}{\sum_1^k w(e_k)}$$

To find the shortest path between m1 and m2, we use bidirectional breadth first search, and we set a threshold k to constraint the number of steps to guarantee performance. If we find no path within k steps, we set struct(m1;m2) = 0.

2.6.2 Pairwise Concept Similarity

A simple pairwise concept similarity is one which considers the shared incoming links in the knowledge base. We call it the Wikilink pairwise similarity, and use the below equation to measure it.

$$ctxt(m_1, m_2) = 1 - \frac{\log(\max(|M_1|, |M_2|)) - \log(|M_1 \cap M_2|)}{\log|W| - \log(\min(|M_1|, |M_2|))}$$

Another novel pairwise similarity measure between two concepts that uses a neural network to represent concepts as continuous vectors is also described.

2.6.2.1 Concept2Vector Representation

We expand upon the Skip-Gram model by Mikolov et al. to represent concepts as continuous vectors. We refer to our method as Concept2Vector. We first describe how we train the model and then how we use it for similarity.

Data Corpus preparation: To train a model for representing concepts as vectors, we need a data corpus that provides the context of each concept. We exploit the Wikipedia internal links for that. An internal link in a Wikipedia page (referred to as wikilink) comprises a reference (hyperlink) to another Wikipedia article and an optional surface form that represents the referenced article. Since each Wikipedia article corresponds to a concept in DBpedia, we consider each Wikilink as a reference to a DBpedia concept. We preprocess all Wikipedia pages, replacing the surface form in each wikilink by the title of the referenced article. If the referenced article redirects to another page then we replace the surface form by the title of the target page. Consider, for example, the wikilink with a surface form *Oxford University* that references the page *University_of_Oxford*, and assume that the concept *Phi_Beta_Kappa* redirects to *Phi_Beta_Kappa_Society*. Then the text "He is an alumnus of *Georgetown University*, where he was a member of *Kappa Kappa Psi* and *Phi Beta Kappa* and earned a *Rhodes Scholarship* to attend the *Oxford University*" is replaced by "He is an alumnus of *Georgetown_University*, where he was a member of *Kappa_Kappa_Psi* and *Phi_Beta_Kappa_Society* and earned a *Rhodes_Scholarship* to attend the *University_of_Oxford*".

After pre-processing, each page in Wikipedia contains original words (that do not appear as part of a wikilink) and the concepts, each treated as one term.

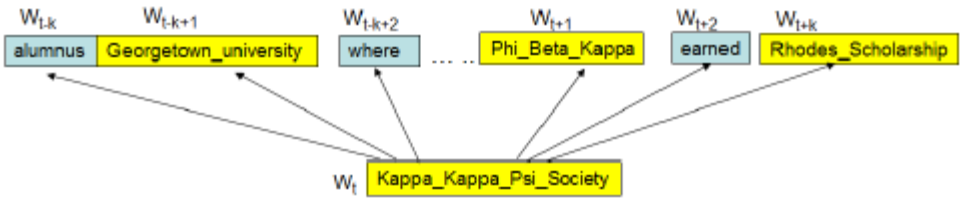

Fig 3: The training example of the sentence

2.6.3 Document Similarity

We extend the groupwise similarity measure of 4.2.2 by incorporating weights into the equation. Let {m1}p be the concepts in the concept graph of D1 and {m2}q the concepts in the concept graph of D2. Let {w1}p and {w2}q be their respective weights. We define the semantic relatedness between D1 and D2 as:

$$ConceptGraphSim(D_1, D_2) =$$
$$0.5 * \frac{\sum_1^p w_{1_i} * best(m_{1_i})}{\sum_1^p w_{1_i}} + 0.5 * \frac{\sum_1^q w_{2_j} * best(m_{2_j})}{\sum_1^q w_{2_j}}$$

where best(m1i) (best(m2j) respectively) is the best pairwise similarity for m1i (m2j respectively) over all concepts in {m2j}q ({m1i}p respectively).

To derive the final similarity score of two documents we use the cosine similarity.

CHAPTER 3

APPLICATIONS

3.1 Theoretical Applications

- Preserves the word ordering and semantic relationships between words.
- Improves upon previous CBOW (Continuous Bag of Words) and Skip-Gram Models.
- Can be restricted in terms of dimensions.

3.2 Practical Applications

- Can be used to generate word embeddings. These embeddings can be used to build intelligent machines based on the hypothesis that 'background knowledge is the key to intelligent decision making'
- Can be used to build large scale recommender systems. These recommender systems are arguably better than Collaborative Filtering Models that suffer from 'cold-start' problem.
- Provides a contextual learning platform which can further be extrapolated to encompass any kind of 'domain-specific' contextual learning in a short period of time.

3.3 Example Application

The doctor/nurse works through his/her patient cohort as part of his/her daily duties. Based upon historical patient records, disease conditions, and treatments ordered, the contextual training platform learns to identify critical to quality search vectors - to index and retrieve (re)training material (newsletters, videos, case studies etc.) from a knowledge platform (e.g. webmd.com). The platform then presents a curated set of training materials for the care provider for self-training. As an enhancement, the contextual training platform could also suggest and connect to in-person contextual expert for high risk cases. The learning inputs can come from standard knowledge source(s) or even from successful care pathways mined from an active information retrieval (IR) system.

CHAPTER 4

CONCLUSION

We presented concept graph for representing a document using its detected concepts. The concept graph utilizes DBPedia to explore relationships among the detected concepts and weighs the concepts according to their closeness centrality to reflect their relevance to the aspects of the document. We then presented a novel similarity measure ConceptGraphSim between two documents by comparing their concept graphs. The similarity measure first represents concepts as continuous vectors using neural network and then combines pairwise similarity between pairs of concepts using their continuous vector representations and their assigned weights. By adding more features, we can improve our results. We further believe that the combination of concept based centrality with neural networks has a high potential that can be further exploited to achieve better results.

For future work, we can add more features and improve the quality of the concept graph representation by either improving the accuracy of the mention detection tool or enriching the concept graph with additional related concepts from the knowledge base.

REFERENCES

[1] G. Salton. The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice - Hall, Upper Saddle River, NJ, USA, 1971.

[2] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Commun. ACM, 1975.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 1990.

[4] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research (JMLR), 2003.

[6] N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In Proceedings of 16th European Conference on Artificial Intelligence, 2004.

[7] M. D. Lee, B. Pincombe, and M. Welsh. An empirical evaluation of models of text document similarity. In Proceedings of Annual Conference of the Cognitive Science Society (CogSci), 2005.

[8] S. P. Borgatti. Centrality and network flow. Social Networks 27, 2005.

[9] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia based explicit semantic analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.

[10] D. Milne and I. H. Witten. An effective, low cost measure of semantic relatedness obtained from wikipedia links. In Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence, 2008.

[11] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. Wikiwalk: Random walks on wikipedia for semantic relatedness. In Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, 2009.

[12] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. Web Semantics, 2009.

[13] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. Artificial Intelligence Research, 2010.

[14] P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. IEEE Software 29(1), 2012.

[15] L. Huang, D. Milne, E. Frank, and I. H. Witten. Learning a concept-based document similarity measure. Journal of the Association for Information Science and Technology, 2012.

[16] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In Proceedings of the 9th International Conference on Semantic Systems (I-Semantics), 2013.

[17] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence, 2013.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Proceeding of the International Conference on Learning Representation (ICLR) Workshop, 2013.

[19] M. Schuhmacher and S. P. Ponzetto. Knowledge-based graph document modeling. In Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM), 2014.