GRAPH-BASED KEYPHRASE EXTRACTION USING WIKIPEDIA

Bharath Dandala, B.Tech.

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

December 2010

APPROVED:

Paul Tarau, Major Professor
Rada Mihalcea, Comittee Member
Miguel E Ruiz, Committee Member
Ian Parberry, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
James D. Meernik, Acting Dean of the Robert B. Toulouse School of Graduate Studies

Dandala, Bharath. Graph-Based Keyphrase Extraction Using Wikipedia. Master of Science (Computer Science), December 2010, 45 pp., 2 tables, 6 figures, 37 references.

Keyphrases describe a document in a coherent and simple way, giving the prospective reader a way to quickly determine whether the document satisfies their information needs. The pervasion of huge amount of information on Web, with only a small amount of documents have keyphrases extracted, there is a definite need to discover automatic keyphrase extraction systems.

Typically, a document written by human develops around one or more general concepts or sub-concepts. These concepts or sub-concepts should be structured and semantically related with each other, so that they can form the meaningful representation of a document. Considering the fact, the phrases or concepts in a document are related to each other, a new approach for keyphrase extraction is introduced that exploits the semantic relations in the document. For measuring the semantic relations between concepts or sub-concepts in the document, I present a comprehensive study aimed at using collaboratively constructed semantic resources like Wikipedia and its link structure.

In particular, I introduce a graph-based keyphrase extraction system that exploits the semantic relations in the document and features such as term frequency. I evaluated the proposed system using novel measures and the results obtained compare favorably with previously published results on established benchmarks.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1. Keyphrase Extraction

Keyphrases are words or phrases that indicate the main topics in a document [12]. Keyphrases describe a document in a coherent and simple way, giving the prospective reader a way to quickly determine whether the document satisfies their information needs. The pervasion of the huge amount of information on World Wide Web, with only a small amount of documents have keyphrase generated, there is a definite need to build keyphrase generation systems. Manual assignment of keyphrase is a tedious and time-consuming task, especially considering the proliferation of the web. Thus, there is a great need for means of automatic keyphrase generation systems. Keyphrase generation is an approach to collect the main topics of a document into a list of phrases. The task of automatic keyphrase generation is divided into two groups: keyphrase assignment and keyphrase extraction. In keyphrase assignment, all potential keyphrases appear in a predefined vocabulary and the task is to classify documents into different keyphrase classes. In keyphrase extraction, keyphrases are available in the document,although authors occasionally supply keyphrases that they do not appear in the document [9]. Automatic keyphrase extraction is the identification of the most important phrases of a document by computers rather than human beings. The automatic keyphrase extraction problem can be viewed as given a bag of phrases for a document, where each phrase belongs in one of two possible classes: either it is a keyphrase or it is a non-keyphrase [34].

1.2. Motivation

The motivation for this research is from mining historical documents. The keyphrase extraction answers a lot of historical questions which leads to a new research direction. The following sections include the research questions that can be answered from historical documents using keyphrase extraction.

1.2.1. Mining Historical Documents:Austin Papers

1.2.1.1. The Stephen F. Austin Papers

The Austin papers are the surviving personal papers and collected documents of Stephen F. Austin. The vast majority of the collection consists of Austin's personal correspondence with hundreds of people in the United States and Mexico during the 1820s and 1830s. Austin spent an enormous amount of time in conversation with both the Americans coming into his colonies and Mexican government officials. Many Mexicans treated Austin as a conduit for communication with all American colonists. Americans, in turn, treated Austin as a source of all information about both Texas and the Mexican government. Austin, as a result, received a veritable flood of letters from both would-be American colonists and Mexican government officials throughout the 1820s and 1830s, which make up the bulk of the Austin Papers collection. These letters document nearly every conceivable aspect of the migration of Americans into Mexico during this period. Potential colonists wrote letters to Austin demanding details on every minute of Texas settlement, including issues of regional infrastructure, attitudes of the Mexican government, agricultural potential of the land, conditions of the territory, perspectives on the local Native American groups, and so on. Austin, in turn, would often answer these letters in painstaking detail, providing documents that contain remarkably detailed portraits of life and conditions in the U.S.-Mexican borderlands on the eve of American conquest. Austin communicated with Mexican officials in equally exacting detail about the movement of these Americans, and his communication with them provides another window into the early government of Mexico and its attitudes toward this in-migration of Americans. While the

majority of the collection consists of documents written in English, significant portions of the Austin Papers were written in Spanish. When Austin arrived in Mexico, he quickly learned the Spanish language and used it as his primary means of communication with Mexican officials. Austin's ability to communicate effectively in both languages further cemented his pivotal role as a central conduit of information between the Mexican government and American colonists (since he could translate between the two), and proved central to much of Austins success as a land agent and colonizer. As such, nearly all of Austins correspondence with Mexican officials was conducted in Spanish and thus a large proportion of the Austin Papers remains in Spanish. In addition to material on the 1820s and 1830s, the papers also contain pre-1821 correspondence of both Stephen F. Austin and his father, Moses Austin, going back as early as the 1780s. There are also legal documents, receipts, and other various miscellaneous documents (such as maps) scattered throughout the collection.

1.2.1.2. Value of Austin Papers to Historians

The historical value of these papers was recognized during Austin's own time, and thus the collection was preserved at the time of his death as an indispensable record of the early nineteenth-century American Southwest. The Austin Papers, as the printed version became known, was originally published in several volumes that covered about 3,000 pages: Eugene C. Barker, The Austin Papers, 2 vols. (Washington, D. C.: Government Printing Office, 1924, 1928). The Austin Papers have, in turn, served as the bedrock for most histories written about Texas during this era, as well as the transition of Far North Mexican into the American Southwest. One of the most consequential of these works was a biography published by Eugene Barker in 1925 (*Eugene C. Barker, The Life of Stephen F. Austin, Founder of Texas, 1793-1836*: "A Chapter in the Westward Movement of the Anglo-American People" [Nashville: Cokebury Press, 1925]) which became the most influential work on how historians during the twentieth century interpreted this period. Another major Austin biography came

out in 1999, published by historian Gregg Cantrell, which helped to further cement the importance of the Austin Papers to historians of the period (Gregg Cantrell, Stephen F. Austin: Empresario of Texas [New Haven: Yale University Press, 1999]). In the interpretations of both Barker and Cantrell, and nearly every other historian, Stephen F. Austin emerges as one of the pivotal players in the transformation of the Mexican Far North into the American Southwest. As such, Austins life and biography (and thus this particular collection of documents) have become the foundation upon which nearly every major historical interpretation of this era stands. As an example, a recently published history of the United States from 1815 to 1848Daniel Howes What Hath God Wrought: The Transformation of America, 18151848 (Oxford, 2007)begins with the role that Stephen F. Austin played during this period. Winner of the Pulitzer Prize for history, Howes account of this era embodies the importance of the Austin Papers collection for our historical understanding of this crucial period in North American history.

1.2.2. Applications of Keyphrase Extraction to Austin Papers

1.2.2.1. Keyphrase Associations over Time

Extracting keyphrases helps in understanding the main topics of the paper quickly. The Austin Papers cover a wide range of topics, and the papers are not domain-specific. Most documents in the Austin Papers contain information identifying the writer of the letter, the receiver of the letter, the date sent, and the place in which the letter was written. Extracting keyphrases associated with these sets of meta-data enables a historian, for example, to examine discussions of important topics (that is, keyphrases) changed over time in the collection of the Austin Papers. So, for example, a topic of great contention during the 1820s in Austin's colony was the matter of slavery, and how it was perceived and addressed by authorities in the Mexican government. Examining how keyphrases associated with a term like "slavery" would enable a historian to examine how discussions concerning this issue evolved over time.

1.2.2.2. Keyphrase Associations with a Person

The Austin Papers contain the name of the person associated with a particular letter. In a similar fashion to the Keyphrase Associations over Time, this would allow a historian to examine the changing concerns and language use of a particular person writing documents in the Austin Papers collection. So a historian, for example, could examine the changing concerns of someone like Stephen F. Austin, particularly over time, by examining the shifts and changes in the keyphrases associated with him. In this example, Austin may well have discussed matters of slavery more often during the 1820s, but shifted to concerns about stability in the Mexican government by the 1830s.

1.2.2.3. Keyphrase Associations between Two Persons

Since the Austin Papers contain information about both the receiver and the writer information of a given letter, a historian could use that as a means of examining the evolution of the discussions (and thus the relationship) between particular people who exchanged letters over the course of a period of time. Austin, for example, maintained correspondence with particular people, like the Mexican representative for Texas, over long periods of time. By extracting the keyphrases of letters between those two people, a historian could use that as a means of examining the evolution of the relationship between those two people.

1.2.2.4. Keyphrase Associations with Geography

Since the Austin Papers contain information about the place in which the letter is written, a historian could use that as a means of evolution of discussions related to geographical location. So a historian, for example, could examine the topics associated with a place, particularly over time, by examining the shift and changes associated with the place.

The above sections clearly explained how single-document keyphrase extraction can help in answering various questions related to historical events. In this thesis, the major concentration is placed on improving the Single-Document keyphrase extraction methods.

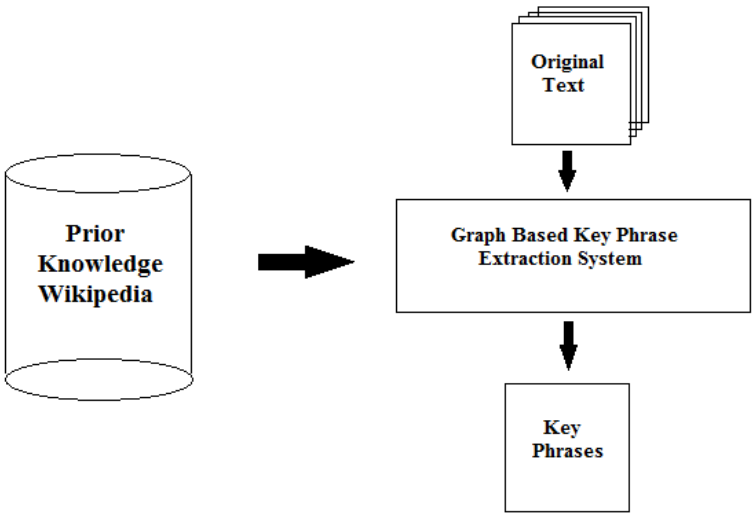

Figure 1.1. Architecture of Keyphrase Extraction

Industry insiders loading up on cheap company stock. A surge of telecom executives and directors purchasing their own companies, stock in the last two months points toward a renewed optimism in the beleaguered sector, say some observers, who view the rash of insider buying as a vote of confidence from management. Airgate PCS, Charter Communications, Cox Communications, Crown Castle International, Nextel Communications and Nortel Networks all have seen infusions of insider investment this summer, echoing trends in both the telecom industry and the national economy.

Figure 1.2. Abstract from the INSPEC Corpora

Telecom Industry
Insider Investment

Figure 1.3. Manually Extracted Keyphrases for the Abstract in Figure 1.2

This thesis focuses on using Wikipedia as a resource to calculate the semantic relatedness between phrases and implementing a method for graph-based keyphrase extraction. We have investigated the role of the semantic graphs in texts and how they can be exploited to identify the keyphrases in the document. The main contribution of this research focuses on ranking the phrases associated with the document using semantic graphs. Finally, we have investigated the approaches for evaluating our proposed keyphrase extraction.

To achieve these goals,the main objectives of this thesis are:

- Wikipedia as a lexical resource
- Semantic Relatedness between phrases
- Extracting keyphrases from a single document
- Evaluation approaches for keyphrase extraction

The general architecture of the system is presented in figure 1.1. Our system uses Wikipedia as prior knowledge and for building semantic graphs and weighted pagerank algorithms for ranking the phrases. Figure 1.2 shows an example article and Figure 1.3 shows keyphrase of a document.

1.4. Thesis Outline

In Chapter 2, necessary background information and related work in keyphrase extraction is outlined. Previous research about semantic relatedness is briefly introduced. It introduces briefly the previous research for graph-based keyphrase extraction. It also introduces the external libraries that are used in this thesis like part-of-speech tagger, chunker, Wikipedia miner briefly.

Chapter 3 introduces the Wikipedia link-based approach for measuring semantic relatedness. It also describes the approach for measuring semantic relatedness between phrases using Hungarian Bipartite Maximum Matching.

Chapter 4 describes about the construction of PhraseGraphs using Wikipedia. It also describes graph-based keyphrase extraction.

Chapter 5 provides the results of approaches with existing approaches. It also compares our results with existing graph-based keyphrase extraction approach, TextRank.

Chapter 6 describes the conclusion and other research directions for future work.

CHAPTER 2

BACKGROUND RESOURCES AND RELATED WORK

2.1. Related Work in Keyphrase Extraction

The work on keyphrase generation can be categorized into two major approaches: extraction and assignment. Keyphrase extraction methods select phrases present in the source document itself. Keyphrase extraction approaches usually consist of a two stages: candidate identification and selection. Candidate identification stage, identifies the candidate phrases in a document. Most of the systems used noun phrases and adjective phrases as candidate phrases for the selection stage. In the selection stage, the phrases that best explains the contents of the document are selected from candidate phrases.

In contrast to keyphrase extraction, keyphrase assignment is typically used when the set of possible keyphrases is limited to a known, fixed set, usually derived from a controlled vocabulary or set of subject headings. From these set of phrases phrases are assigned to a document using various approaches. This problem can be viewed as a multi-class text classification problem.

The attempts on solving the problem of keyphrase extraction can be classified into two main ways: Supervised and Unsupervised

2.1.1. Supervised Keyphrase Extraction

A machine learning technique in which a system uses a set of training examples to learn and perform the required task correctly. Supervised approaches are introduced in keyphrase extraction in which the system learns a keyphrase extraction model that is able to classify candidates as keyphrases. KEA[37] is an automatic keyphrase extraction algorithm based on a domain-specific machine-learning model [Frank et al., 1999]. The system is trained using

two features TFIDF (Salton, 1988) and Distance (the ratio of the number of words before the first appearance of the phrase in the document and the total number of words in the document). Then, the Naive Bayes classifier is used to extract the potential keyphrases from a document. Hulth[13][14] adds more linguistic knowledge, such as syntactic features which significantly improved the performance. A number of supervised approaches are introduced to improve KEA,adding more lexical and syntactic features[35] [26]

2.1.2. Unsupervised Keyphrase Extraction

Unsupervised approaches involve two steps: candidate selection and ranking. In candidate selection step a set of phrases from the document are selected. In ranking phase, the candidates phrases are ranked and the top n phrases are determined as keyphrases. Noun phrases are selected as keyphrases in many unsupervised approaches in keyphrase extraction[2] [5] Length of the phrase, term frequency, head noun frequency and other term features are used to rank the keyphrases in ranking phase. Graph-based keyphrase extraction was proposed (Mihalcea and Tarau, 2004) in which the tokens are nodes and edges are co-occurrence relations between the tokens in a document. PageRank(Page et al., 1999) was used in the ranking stage which is known as TextRank [22]. Finally, from the top n-tokens, tokens are merged if they are adjacent in the document. Several variation of TextRank[36] [11] are experimented to improve the performance of TextRank. Lexical semantic graphs[33] are introduced in which tokens are nodes and edges represent the semantic relatedness between the tokens in the document. PageRank[27] was used in the ranking stage and from the top n-tokens, tokens are merged if they are adjacent in the document.

2.2. Linguistic Background

A document is not just a bag of sentences, but it has grammatical structure and meaningful data. Linguistic semantics is the study of meaning as conveyed through language. It typically focuses at the level of words, phrases, sentences and larger units of discourse. The basic area of study is the meaning of signs, and the study of relations between them. The

relations are homonymy, synonymy, antonymy, polysemy, paronyms, hypernymy, hyponymy, meronymy, metonymy, holonymy, linguistic compounds. A key concern is to bring a meaningful representation of large chunks of text, using the composition from smaller units of meaning. The documents written by a human are expressions of intentional and mindful activity. Thus, The document evolves around one or more general concepts or sub-concepts. These concepts or sub-concepts should be structured and semantically related to each other, so that they can form up the meaning representation of a document. Considering the fact, the phrases or concepts in the document are related to each other, a new method for keyphrase extraction exploiting the semantic relations is introduced in this thesis.

2.2.1. Semantic Resources

Semantic resources are knowledge bases containing words (or concepts) and relationships between them. Wordnet[7] is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. However, it is very expensive and time-consuming to create such resources, and they usually cover only a limited number of relations. Thus, in this thesis we investigate the applicability of other semantic resources that are collaboratively constructed on the Web like Wikipedia. Wikipedia is a multilingual, Web based, freely available encyclopedia, constructed in a collaborative effort of voluntary contributors and growing exceptionally. As Wikipedia is freely available and quickly growing, it constitutes a possible substitute for Wordnet. In this thesis, we are going to investigate the applicability of Wikipedia for computing the semantic relatedness between phrases.

2.2.2. Semantic Relatedness

Semantic relatedness is an important concept used in a variety of applications,such as information retrieval, automatic indexing, word sense disambiguation, keyphrase extraction

and automatic summarization. It provides the ability to measure the relatedness of two concepts. Semantic similarity and semantic relatedness are sometimes used interchangeable in the literature. These terms however, are not identical. Semantic relatedness indicates degree to which words are associated via any type (such as synonymy, meronymy, hyponymy, hypernymy, functional,associative and other types) of semantic relationships. Semantic similarity is a special case of relatedness and takes into consideration only hyponymy/hypernymy relations. The relatedness measures may use a combination of the relationships existing between words depending on the context or their importance. Humans can easily judge the semantic relatedness between two words using their previous experience and knowledge. For example, a person can easily say that "Honey" and "Bee" have a stronger relationship compared to "Food" and "Football". For a machine to solve this problem it needs large datasets or semantic resources like Wordnet or Wikipedia.

2.3. Collaborative Resources and Knowledge Bases

In this section, the collaborative and knowledge bases that are used in this thesis are discussed. The resources and knowledge bases that are discussed in this section are:

- Wikipedia
- Wikipedia Miner

2.3.1. Wikipedia

Wikipedia is a Web based, freely available encyclopedia, constructed in a collaborative effort by various participants. In the recent years, the potential of Wikipedia attracted many researchers to explore it in the field of Natural Language Processing. It has been used in NLP tasks like text categorization (Gabrilovich and Markovitch, 2006), information extraction (Ruiz-Casado et al., 2005), information retrieval (Gurevych et al., 2007), question answering (Ahn et al., 2004), computing semantic relatedness (David Milne et al.,2008), or named entity recognition (Bunescu and Pasca, 2006). The statistical natural language processing (NLP)

highly relies on the size and recency of the corpora. Currently, the English Wikipedia alone has over 3,433,568 articles of any length. The combined Wikipedias for all other languages greatly exceeded the English Wikipedia in size, giving a combined total of more than 1.74 billion words in 9.25 million articles in approximately 250 languages(Wikipedia Size_of_Wikipedia). Wikipedia corpora is rich in hyperlinks and its hyperlink structure draws a huge directed graph which helps in identifying the relationship between concepts. The majority of these concepts or links correspond to entities, which are related to another concept in Wikipedia being described, and have a separate entry in Wikipedia. Wikipedia provides millions of these links through which Wikipedia can be represented as a large semantic network. This large semantic network helps in calculating semantic relatedness between two concepts or entities. The main characteristics of Wikipedia are Disambiguation of Articles by URL, the Link Structure, Anchor Texts and Domain-specific senses.

2.3.1.1. Disambiguation of Articles

Word sense disambiguation by URL is one of the most notable characteristics of Wikipedia. For example, if a sentence "A bank is a financial intermediary" exists in an article in a dictionary, human understand that the word bank is a "financial bank" instead of "river bank". In Wikipedia, almost every article corresponds to exactly one concept and has its own URL respectively. Both "financial bank" and "river bank" have their own Wikipedia pages. If a human is writing another article and encounters the word "bank", then a corresponding article can be linked to the word "bank" based on the context of the article.

2.3.1.2. Article Links

In Wikipedia the links between articles show association between concepts of articles. Hence they can be used to find related concepts for an article. The type and strength of the relation is not represented in Wikipedia. The representation of type and strength would make Wikipedia a large semantic net, which helps in solving a lot of Natural Language Processing problems.

2.3.1.3. Wikipedia Category Graph

Articles in Wikipedia are organized into a hierarchy. The articles are organized in a way, such that an arbitrary number of articles belong to a particular category. Each particular category is divided into one or more sub-categories. Thus the categories in Wikipedia form a large semantic taxonomy like Wordnet.

2.3.1.4. Wikipedia Link Structure Mining

By analyzing the category graph or Wikipedia link structure various problems can be solved like link structure analysis of Wikipedia which can be used in calculating semantic relatedness. It also provides an evidence, if two pages are sharing similar links then they have topically similar content compared to pages that are not sharing similar links.

2.3.2. Wikipedia Miner

Wikipedia Miner[24] is a toolkit for using the structure, content and understanding the relations between entities of Wikipedia. It mainly aims at providing a easy way to integrate Wikipedia's knowledge into our own applications, by providing simplified, object-oriented access to Wikipedia's structure and content. It helps in understanding and measuring the semantic relatedness between terms or concepts in Wikipedia. It also helps in detecting and disambiguating Wikipedia topics when they are mentioned in documents. Wikipedia Miner exploits the links structure of Wikipedia to calculate the semantic relatedness between two concepts. In this thesis, the Wikipedia Link-Based Measure [25] is used for calculating the semantic relatedness between two concepts.

2.4. External Libraries

The external libraries that are used in this thesis are discussed in this section:

- Part-of-Speech Tagger
- Chunker

2.4.1. Part-of-speech tagging

Part-of-speech tagging , also called grammatical tagging or word-category disambiguation, is the process of assigning the words in a text document as corresponding to a particular part-of-speech, based on both its definition, as well as its context. Traditional grammar classifies words based on eight parts of speech: the verb, the noun, the pronoun, the adjective, the adverb, the preposition, the conjunction, and the interjection. Part-of-speech tagging is an essential tool in many natural language processing applications such as information extraction, word sense disambiguation, parsing, question answering, and named entity recognition. Manually assigning part-of-speech tags to words is a tedious and time-consuming task. Automatic Part-of-speech tagging is a process of assigning part-of-speech by a computer instead of doing it manually. Contextual behavior of the words largely vary in different languages, so the key issue is to identify the part-of-speech of a word based on the context in which it occurred. There are several approaches to automatic part-of-speech tagging - rule based, probabilistic, and transformational-based approaches. Probabilistic approaches determine the most probable tag of a token based on the surrounding context words,based on probability values obtained from a training corpus. The most widely known probabilistic approaches for tagging are Hidden Markov Models, Maximum Entropy Markov Models and Conditional Random Fields(CRF).

2.4.1.1. Conditional Random Fields Tagging

Conditional random fields (CRFs) [17] is a framework for building probabilistic models to segment and label sequence data. CRFs offer several advantages over hidden Markov models (HMMs) and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models. CRFs also avoid a fundamental limitation of maximum entropy Markov models (MEMMs) and other discriminative Markov models based on directed graphical models, which can be biased towards states with few successor states. A CRF is an undirected graphical model for labeling sequential data. CRF tagger has been

applied to English [29] with an accuracy of 97%. so, in this thesis, a CRF Tagger has been used to perform the part-of-speech tagging on English Documents.

2.4.2. Chunking

Phrase chunking is a process of dividing sentences into non-overlapping phrases or its sub constituents. A robust chunker or shallow parser has become an essential tool in a variety of NLP applications such as Information Extraction, Question and Answering and Information Retrieval. The linguistic rule based chunkers are fragile because of special cases occurring in the language and need a relatively long time to develop the rules. A chunker or shallow parser identifies simple or non-recursive noun phrases, verb groups and simple adjectival and adverbial phrases in a corpus.

2.4.2.1. IOB Chunking

Chunk structures can be represented using either tags or trees. The most widespread file representation uses IOB tags. In this scheme, each token is tagged with one of three special chunk tags, Inside(I), Outside(O), Begin(B). A token is tagged as B if it occurs at the beginning of a chunk. Subsequent tokens of the same chunk are tagged I. The remaining tokens are tagged O. CRF chunking is best suited for labelling sequential data. The CRF models are trained on the feature templates for predicting the chunking boundary. Finally the chunk tags are merged to obtain the appropriate chunks. CRF chunker has been applied to English [28] with an accuracy of 95.77%.

2.5. Chapter Summary

In this chapter, we introduced the problem of keyphrase generation and different categories of keyphrase generation: keyphrase extraction and keyphrase assignment. We explained various types of keyphrase extraction methods and previous approaches for automatic keyphrase extraction. We briefly introduced the linguistic background, external libraries and knowledge bases of this thesis.

CHAPTER 3

SEMANTIC RELATEDNESS

Semantic relatedness(SREL) is a measure of strength of relation between two concepts or terms. A value between 0 and 1 is used to express the strength of relation.

(1) $$SREL(c1, c2) \in [0, 1]$$

3.1. Semantic Relatedness Measures

The existing measures for determining semantic relatedness can be classified into four broad categories.

3.1.1. Path-based Measures

Path-based measures compute relatedness as a function of the number of edges between the two concepts the words are mapped to. Semantic relatedness is represented as an inverse of this distance function between the concepts the words mapped[30]. Leacock and Chodorow (LC)[18] proposed a normalized distance measure which takes into account the depth of the taxonomy in which concepts are found.

(2) $$LC_{sim}(c1, c2) = -log\frac{length(c1, c2) + 1}{2 * depth}$$

Wu and Palmer (1994)(WUP) proposed a method which takes into account the depth of the nodes together with the depth of their least common subsumer.

(3) $$WUP_{sim}(c1, c2) = \frac{2 * depth(lcs)}{2 * depth(lcs) + length(c1, lcs) + length(c2, lcs)}$$

3.1.2. Information Content based Measures

Information Content is a measure of specificity for a concept. The more specific the concept gets, the more Information it has. Information content is a measure based on frequency count of the concepts. For a concept c, information content can be defined as the negative logarithm of the probability of that concept. The probability p(c) can be estimated from the relative corpus frequency of c and the probabilities of all concepts that c subsumes (Resnik, 1995)

$$IC(c) = -\log(p(c)) \tag{4}$$

Information content approaches for relatedness are based on the assumption that the amount of information that two concepts share determines their relationship. There are three Information Content measures implemented by Resnik[31],Jiang and Conrath (JCN)[16], and Lin (LIN)[20]. All these measures rely on the idea of a least common subsumer (LCS). A least common subsumer is the most specific concept that is a shared ancestor of the two concepts in a taxonomy. The measure of Resnik(RES) computes the similarity as a function of their information content of LCS of the two concepts.

$$RES_{sim}(c1, c2) = IC(LCS(c1, c2)) \tag{5}$$

Both the Lin(LIN) and Jiang & Conrath(JCN) measures attempt to refine the Resnik measure by augmenting it with the Information Content of the individual concepts being measured in two different ways:

$$LIN_{sim}(c1, c2) = \frac{2 * res(c1, c2)}{IC(c1) + IC(c2)} \tag{6}$$

$$JCN_{sim}(c1, c2) = IC(c1) + IC(c2) - 2 * res(c1, c2) \tag{7}$$

3.1.3. Gloss Based Measures

A gloss is a brief notation of the meaning of a word or wording in a text. Large taxonomies like Wordnet usually contain short glosses for each concept. Gloss overlaps were introduced by Lesk (LES)[19] to perform word sense disambiguation. A measure (LES) based on the amount of word overlap in the glosses of two concepts is used to measure the relatedness between two concepts.

$$LES_{sim}(c1, c2) = \|gloss(c1) \cap gloss(c2)\| \tag{8}$$

where gloss(ci) returns the multi-set of words in a concept gloss. An extended Gloss overlap measure was proposed by Banerjee and Pederson (BPE)[1] that extends the glosses of the concepts to include the glosses of the other concepts to which they are related.

$$BPE_{sim}(c1, c2) = |extGloss(c1) \cap extGloss(c2)| \tag{9}$$

where extGloss(ci) returns the multi-set of content words in the extended gloss.

3.1.4. Vector-Based Measures

Patwardhan and Pedersen (2006) proposed another gloss-based semantic relatedness measured by creating aggregate co-occurrence vectors for a Wordnet sense by adding the co-occurrence vectors of the words in its Wordnet gloss. The distance between two senses is then determined by the cosine of the angle between their aggregate vectors. Another variant of vector-based approached is proposed by Gabrilovich and Markovitch(GM)[10] where the meaning of a word w is represented as a high- dimensional concept vector. Each element represents the document and the value of the element depends on whether the word w occurs in that particular document. If the word does not exist in the document then the element is 0 else the value is term frequency inverse-document frequency(TFIDF) of the word. The final similarity is calculated by using cosine similarity between the highdimensional concept

vectors.

$$GM_{sim}(w1, w2) = \frac{\vec{d}(w1) * \vec{d}(w2)}{|\vec{d}(w1)||\vec{d}(w2)|} \tag{10}$$

where $\vec{d}(wi)$ is the high dimensional context vector of the word wi.

Wikipedia link structure was exploited by Milne and Witten to introduce a Wikipedia link-based measure. The Wikipedia link-based measure takes the advantage of heavy linking between articles in Wikipedia. In this thesis, we are adapting Wikipedia link-based measure for measuring semantic relatedness between phrases. In the next sections, Wikipedia link-based measure and adapting it to measure semantic relatedness between phrases are discussed.

3.2. Semantic Relatedness using Wikipedia

Semantic Relatedness between terms using Wikipedia was first introduced by Strube and Ponzetto and it is known as WikiRelate[32]. WikiRelate exploited the category network structure for calculating path based measures that are mostly used on Wordnet. Explicit Semantic Analysis method was proposed by Gabrilovich and Markovitch, in which the meaning of the texts is represented by high-dimensional space of concepts derived from Wikipedia. Machine learning techniques were used to build a semantic interpreter that maps fragments of natural language text into a weighted sequence of Wikipedia concepts ordered by their relevance to the input. The input texts are represented as weighted vectors of concepts, called interpretation vectors. To obtain the relatedness score, the cosine similarity is computed between the interpretation vectors.

Milne(MIL)[23] introduced Wikipedia link-vector model that is specific to Wikipedia as it relies on dense linking between articles that cannot be found in other semantic resources. The more the two articles share links, the higher their semantic relatedness. Wikipedia articles related to a particular term are found by first identifying all pages whose titles match the term and process them. In the processing step articles were used directly, redirects were followed so

that their corresponding articles are used, disambiguation pages were processed so that every article that they link to is used. The vector is represented using link counts weighted by the probability of each link occurring in the source document. The weight of a link in the source document is the number of times the source document contains that link multiplied by the inverse probability of any link to the target document. Formally, the weight w of a link between a source document s and a target document t is defined as:

$$w(s \rightarrow t) = |s \rightarrow t| * \log\left(\sum_{i=1}^{N} \frac{N}{|i \rightarrow t|}\right) \tag{11}$$

where N is the total number of documents in Wikipedia.

The source documents are represented as a link weighted vectors v. The semantic relatedness between two source documents is computed as the cosine of the link weight vectors.

$$MIL_{srel}(a1, a2) = \frac{\vec{v}(a1) * \vec{v}(a2)}{|\vec{v}(a1)||\vec{v}(a2)|} \tag{12}$$

Wikipedia link-vector model is refined and a new measure is introduced using the Google Distance[6] known as Wikipedia link-based measure(WLM)[25]. This measure considers both in-coming and out-going links of the source document in Wikipedia. Formally, the semantic relatedness between two articles is defined as:

$$WLM_{srel}(a, b) = \frac{\log(max(|A|, |B|)) - log(|A \cap B|)}{log(N) - log(min(|A|, |B|))} \tag{13}$$

where a and b are the two source documents, A and B are the sets of all articles that link to a and b respectively, and N is the total Number of documents in Wikipedia. In the thesis, Wikipedia link-based measure is adapted to calculate the semantic relatedness between phrases. The method used to measure semantic relatedness between phrases is discussed in the next section.

3.3. Wikipedia Link-based Measure for phrases

The Wikipedia link-based measure is adapted to measure semantic relatedness between phrases. The steps that are involved in measuring the semantic relatedness between phrases. Candidate Article Identification Semantic Relatedness Between Terms Bipartite Weighted Maximum Matching Between Phrases

3.3.1. Candidate Article Identification

The first step in measuring the relatedness between two phrases is to identify the terms of each phrase. This term extraction phase includes matching the phrase with Wikipedia's titles and finding the longest match of the phrase. Doing this process, recursively produces one or more terms. For example, consider the phrase "nextel communications and nortel networks", the first longest match for this phrase in Wikipedia is "nextel communications". The next longest match for this phrase is "nortel networks". So, this phrase has two terms. The stop words are not removed if they occur with another concept and they both combinedly exist as Wikipedia title. For, example consider "United States of America" this phrase is not split as it exists as a Wikipedia title. Then, the semantic relatedness between terms from one phrase and other are measured. The first step in measuring the relatedness between two terms is to identify the Wikipedia articles the articles which discuss them. There are two major problems in this: polysemy and synonymy. Synonyms are words with almost identical or nearly similar meanings. Terms that are synonyms are said to be synonymous, and the state of being a synonym is called synonymy. For example a concept "student" may be referred to as "pupil". Polysemy is the tendency for a concept to have multiple meanings. For example student might refer to a student, or student(newspaper). The correct sense depends on the term which it is compared to, for measuring relatedness. Wikipedia's link structure provides a large number of anchor texts that provides both polysemy and synonymy. The anchor texts in Wikipedia are used to identify candidate articles for terms. For example, when measuring

relatedness between "student" and "school", the "student" article is considered, but not student(newspaper) article.

3.3.2. Measuring Relatedness Between Terms

The candidate article step selects one or more articles for a term, that are synonymous and polysomous. From the obtained articles for each term, similarity between all of their representative articles is measured. Each article corresponding to a term is taken and semantic relatedness is measured with all the representative articles of the other term, one at a time. The Google Distance Measure introduced in section 3.2 is used to measure to measure the semantic relatedness between representative articles. Once this process of measuring relatedness is complete, the ambiguity is resolved to represent or choose a particular article for the term from the candidate articles. The commonness of a sense or article is defined by the number of times it is used as a destination in Wikipedia. For example, the word "student" has more links from other articles in Wikipedia compared to "student(newspaper)". So, the commonness of word "student" is higher than "student(newspaper)". The ambiguity is resolved by combining the commonness and relatedness measure between articles. In the next section, we adapted this approach to calculate semantic relatedness between phrases.

3.3.3. Measuring Relatedness Between Phrases

After the candidate article selection, measuring relatedness between terms, we have term pair semantic relatedness. The simplest way to measure the semantic relatedness between phrases is to take a one-one term mapping between the two phrases and add them up to obtain a semantic relatedness score. But, a more advanced method is introduced in this thesis, which is similar to finding a maximum weight matching in a bipartite graph. Kuhn-Munkres algorithm[8] is implemented to find the maximum weight matching in a bipartite graph. In our problem, we have rectangular matrices so a modified version of Kuhn-Munkres algorithm for rectangular matrices is used[4]. This approach is demonstrated by considering two example

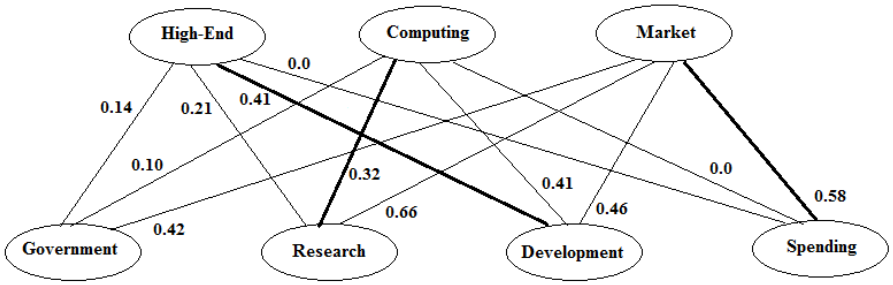

Figure 3.1. Weighted Bipartite Graph

phrases "high-end computing market" and "government research and development spending" In the Candidate Article Selection step, the phrases are segmented as explained in the section 3.3.1. After this process, the phrase "high-end computing market" is segmented into terms high-end, computing and market. Similarly the "government research and development spending" is split into government, research, development and spending. In the next step, semantic relatedness between terms is obtained as explained in section 3.3.2.

A bipartite graph is constructed using the terms and their semantic relatedness as shown in Figure 3.1. The bipartite maximum matching is computed using the Kuhn-Munkres algorithm. The maximum matching for the graph is represented in bold lines in Figure 3.1. Once the matching is obtained, we define a semantic relatedness between the phrases as:

$$SREL_{PHRASE}(Phrase1, Phrase2) = \frac{2 * M}{|A| + |B|} \tag{14}$$

Where M is the sum of weights on the weighted Maximum Matching, $|A|$ and $|B|$ are number of terms in the phrase1 and phrase2.

The semantic relatedness for our example is calculated using the Equation 14.

$$SREL_{PHRASE}Phrase1, Phrase2 = \frac{2 * (0.41 + 0.32 + 0.58)}{3 + 4} = \frac{2.62}{7} = 0.374 \quad (15)$$

3.4. Chapter Summary

In this chapter, we introduced various measures of semantic relatedness. We also introduced semantic relatedness using Wikipedia and how it can be incorporated in our thesis to measure semantic relatedness between phrases.

CHAPTER 4

GRAPH-BASED KEYPHRASE EXTRACTION

In this chapter, we introduce the implementation of graph-based keyphrase extraction system using Wikipedia. This chapter also provides details about graph construction and various approaches of graph-based keyphrase extraction experimented in this thesis.

4.1. Phrasegraphs

Graph-based extraction for keyphrase extraction was first proposed in an approach called TextRank[22]. In the TextRank algorithm, graph nodes are the tokens (nouns and adjectives) and edges represent the co-occurrence relations between the tokens. The pagerank (page et al., 1999) is applied on the graph to retrieve the top n words (where n is the one-third of the length of the document). From the obtained n-words, a further post-processing step of merging adjacent tokens in the original document produces a set of keyphrases. Though, its a new way of looking a problem the TextRank intuitively takes the term frequencies as primary evidence as the term with higher frequency has higher links. Even a term that occurs only once might have higher importance than terms that occur several times. For example, consider a document about a named entity("Obama") which is mostly replaced by pronoun("he") in the document. To resolve these kind of issues, we are introducing phrasegraphs. A phrasegraph is a connected undirected weighted graph, in which nodes represent the phrases in the document and edges represent the semantic relatedness between them. The semantic relatedness between the nodes is measured using the approach introduced in 3.3.3. In Figure 4.1, we showed the phrasegraph for the article shown in Figure 1.2. We represented only those edges that have semantic relatedness or weight greater than 0.2.

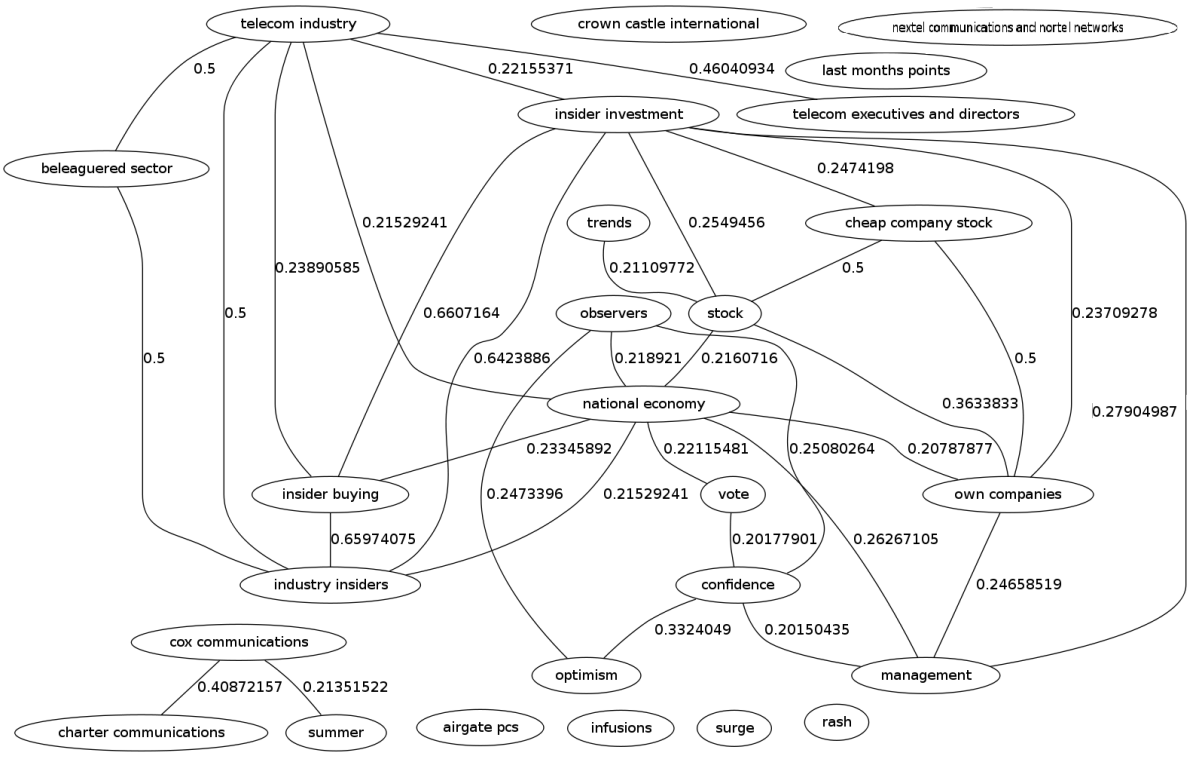

Figure 4.1. PhraseGraph for the Article shown in Figure 1.2

4.2. Keyphrase Extraction Architecture

4.2.1. Pre-processing and Candidate Selection

In the first step of pre-processing, the documents are tokenized and split into sentences. The next steps of pre-processing include Part-of-speech tagging and Chunking. Conditional Random Field(CRF) POS Tagger is employed on the dataset to tag the data set. The Conditional Random Field Chunker is employed on the resulting data to chunk the documents. From this pre-processed set, Noun Phrases and Adjective Phrases are selected as candidates. The extracted phrases are further processed to remove the trailing stop words and words that exactly match the stopword. If the stop words occur in between of a phrase, then they are not removed. For example, "the Dallas morning news" is converted to "Dallas morning news" but keeping "Player of the Year" as it appeared. A further processing is made on the

documents to extract Noun or Adjective tokens that are not part of Noun Phrase or Adjective phrase. The phrases obtained are selected as candidates for keyphrases.

4.2.2. Graph Representation and Phrase-Ranking

The phrases obtained from the previous step are used to construct the phrasegraphs. The graph is represented as a phrasegraph where each node represents a phrase obtained after preprocessing and candidate selection and edge represents the strength or relatedness (or semantic relatedness) between the phrases which is between 0 and 1. After the graph representation step, different types of phrase-ranking approaches have been tried.

4.2.2.1. Pagerank

Pagerank is developed at Stanford University by Sergey Brin and Larry Page and first introduced in [27].It is a web ranking system based on the link structure of the Web. The main idea is that web pages are important, if linked to by important web pages. The main idea behind pagerank is, if a page gets linked from a large number of other pages, then that page is considered important. The importance of the target page also depends on the importance of the pages linking it. The formulation of pagerank is simple.

Let us consider the hyperlink structure of the web as a directed graph G = (V,E) where V is the set of vertices and E is the set of directed edges. The vertices of the graph are the web pages represented as 1,2.....n. The directed edges of the graph represent the hyperlinks from one page to another i.e., if a page i has a link to j then $(i, j) \in E$. Let us represent this graph by its connectivity matrix G, i.e. $Gi, j = 1$ if there is a link from page j to page i else 0.

Let P_j denote the pagerank of vertex j, then

$$P_j = \sum_{j,i \in E} \frac{1}{Out(i)} P_i \tag{16}$$

where Out(i) is out-degree of page i, and i is a page that has link to j.

The process is iterative and stopped when a convergence is achieved. The convergence is tested representing the pagerank problem as a Markov Chain Model.

Pagerank as Markov-Chain Model: The network of web pages are represented as a Markov Chain, in which each state is a page and the connectivity of the graph G is represented by a $n \times n$ transition probability matrix P where $P_{j,i}$ is the probability of entering state j given the current state i. A matrix P is called column stochastic, if $\sum_{j=1}^{n} P_{j,i} = 1$ for all $i \in (1, ..., n)$. The matrix P is called irreducible if $P_{j,i} \in (0, 1)$. The pagerank Equation can be re-written as a Markov Chain model using the matrix P

$$P_{j,i} = G_{i,j}/c_i \tag{17}$$

where $G_{i,j}$ is equal to 1,if there is a link from j to i else 0, c_i is the number of outgoing links from j, if there is a link from page i to page j.

However, there is still one problem with this model because of the web pages which have no outgoing links, so called dangling pages. Because of the dangling pages, the transition probability matrix is not a stochastic matrix. We overcome this problem by adding a link from this page to every other page. i.e. $P_{j,i} = 1/n$ for all j, if i is a dangling page. Now P is a stochastic matrix, but it is still not irreducible in general. To render P irreducible we add the possibility to jump to a random web page, ignoring the hyperlink structure of the web. We assume the probability of this is $(1 - \alpha)$ where $\alpha \in (0, 1)$ which is known as damping factor(usually $\alpha = 0.85$). The transition probability matrix using Markov Chain Model can be represented as:

$$P = \alpha(\hat{G} + \vec{d}u^T) + (1 - \alpha)1v^T) \tag{18}$$

where $\hat{G}$ is the normalized matrix, $\vec{d}$ the characteristic vector of dangling nodes, α is the damping factor and v is the personalization vector which can be used to prefer certain pages and u is the dangling node distribution vector.

4.2.3. Weighted Pagerank

The pagerank algorithm in section 4.2.2.1 for bringing order to web is assuming unweighted graphs. As the phrasegraphs are weighted graphs, with weights representing the semantic relatedness, weighted pagerank algorithm is used. Mihalcea and Tarau[22] introduced a variant of pagerank for weighted graphs, that takes edge weights into account when computing the score associated with a vertex in the graph.

$$WS(V_i) = (1-d) + d * \sum_{v_j \in In(V_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{jk}} WS(V_j) \tag{19}$$

4.2.4. PhraseRanks

In our case, pagerank is the limit distribution of a stochastic process whose states are phrases. The row-normalized matrix of a weighted graph G is the matrix G such that G_{ij} is weight associated between two phrases i and j over the total semantic relatedness associated with i if there is an arc from i to j in G.

Let P be the row-normalized matrix of G,

Let us define d as the characteristic vector of the dangling nodes.

The characteristic vector d can be represented as a vector with one in position corresponding to nodes without outgoing arcs and 0 for the nodes with outgoing arcs.

Let v and u be distributions, which we call the preference and the dangling-node distribution.

Let v be the preference vector based on phrase frequencies or term frequencies or TFIDF scores of a phrase, which significantly conditions the pagerank and,

Let u be the dangling preference on the nodes which significantly conditions the pagerank, then the weighted pagerank can be represented by an Eigen vector equation,

$$x(t+1)' = x(t)'(\alpha P + \alpha \vec{d} u' + (1-\alpha) 1 v') \tag{20}$$

The above eigenvector can be used in three ways.

The first representation is traditional weighted pagerank in which the preference vector is uniform.

The non-uniform preference vector is used to bias pagerank with respect to a selected set of trusted pages, which is introduced in personalized pagerank[15].

Another variation of personalized pagerank was introduced in [3], in which the dangling-node distribution is also non-uniform.

In this thesis, the following three variations of pagerank are experimented with phrasegraphs.

- The traditional weighted pagerank in which both preference vector and dangling-node distribution is uniform.
- The second representation is weakly preferential in which the dangling nodes have a uniform transition towards all the other nodes and v is the non-uniform preference vector($u \neq v$).
- The third representation is strongly preferential in which the dangling nodes have the transition following a preference vector($u = v$).

4.2.4.1. Uniform Phrasegraphs

After the graph is constructed, the score associated with each vertex is set to an initial value of 1/n where n is the number of phrases in the document. The equations 20 and 19 are used to run a weighted pagerank on the graph for several iterations until it converges.

4.2.4.2. Biased Phrasegraphs

In the biased phrasegraphs, two variations of pagerank: "strongly preferential" and "weakly preferential" are tried. In this thesis, we tried three types of preference vectors. In the first approach, the preference vector v is set, which significantly conditions the pagerank using the phrase frequencies. The preference of the phrase i is represented as $Count_i/Countall_{phrases}$. In the second approach, the preference vector v is set, using the phrase weights. In this case,

the phrase weights are calculated using the frequencies of terms in the phrase. Important information is uncovered by looking at repeated terms in a document. The terms are scored based on their recurrence in the corpus. The term scores and length of the phrase are used to determine the weight of a phrase. The phrase weights are calculated using the equation 21 [21].

$$W(P,i) = \beta(n) + \frac{\sum_{k=1}^{n} \theta(ik) tf}{n} \tag{21}$$

where P is the phrase, i is the document, tf is the term frequency in the document, β is the bias based on the length of the phrase and n is the number of terms in phrase. We are considering θ equal to 1.

Term frequency inverse document frequency(TFIDF) is a statistical measure that determines how important a word is to an article in a given corpus. It works by determining the relative frequency of a word in a specific document compared to the inverse proportion of that word over the entire document corpus. The third approach is similar to second approach. But the phrase weights are calculated using the TFIDF scores of the terms in the phrase. As, our experiments are on single-document keyphrase extraction, we used the IDF scores from an external corpus. In this thesis, the IDF scores are from British National Corpus. The phrase weights are calculated using the below equation

$$W(P,i) = \beta(n) + \frac{\sum_{k=1}^{n} \theta(ik) TFIDF_{k,BNC}}{n} \tag{22}$$

where $IDF_{k,BNC}$ is IDF of the term k in BNC corpus. The IDF scores are smoothened to assign a score for those words that are not in BNC corpus. These three approaches are tried with both strongly and weakly preferential as explained in 4.2.4

4.3. Chapter Summary

In this chapter, we introduced the architecture of the keyphrase extraction system. We also explained the differences between pagerank and weighted pagerank. We also introduced the different variations of pagerank tried on phrasegraphs in this thesis.

CHAPTER 5

EVALUATION AND EXPERIMENTAL RESULTS

5.1. Evaluation

Evaluation of the proposed automatic keyphrase extraction system is important for proving the central hypothesis of this thesis. This evaluation answers which is the best of the proposed approaches and how the proposed approaches improved when compared with other graph-based keyphrase extraction algorithms. The most common and recent approaches for evaluating keyphrase extraction algorithms are manual evaluation (Barker and Cornacchia, 2000; Turney, 2000; Jones and Paytner, 2002), automated evaluation against human assigned keyphrases (Frank et al., 1999; Turney, 2003; Hulth, 2003; Mihalcea and Tarau, 2004; Nguyen and Kan, 2007; Torsten,2009).

5.1.1. Manual Evaluation

Human evaluation are subject to specific guidelines given to the human judges when performing the evaluation task. Human judges decide how well the returned keyphrases describe the information in the document. In manual evaluation, the variation of evaluation is highly influenced by the guidelines given to the human assessors.

5.1.2. Automated Evaluation

A gold standard is a result which is pre-defined by the human judges. Automated evaluation is evaluating the performance of the keyphrase extraction system against human annotated gold standard keyphrases. Automated Evaluation is used in evaluating the performance of our keyphrase extraction system as it avoids the problems with manual evaluation.

5.1.3. INSPEC Dataset

The INSPEC dataset contains 2000 abstracts of the journal papers from computer science and information technology. Each abstract contains two sets of keyphrases: controlled keywords, restricted to the thesaurus and useful for keyphrase assignment and uncontrolled terms useful for keyphrase extraction. The data set used in this experiments is 500 abstracts from the INSPEC collection which is same as the dataset used in keyphrase extraction systems reported in (TextRank, 2004 and Hulth, 2003).

5.1.4. Evaluation Metric

Comparing automatically extracted phrases to phrases assigned by human judges is a simple and fast evaluation metric. For each document human judges manually assign the keyphrases based on specific guidelines known as gold standard. The performance of the system is determined using the proportion of phrases extracted by the automatic keyphrase extraction system compared to gold standard(precision) and proportion of the identified phrases out of all possible correct phrases (recall). The F-measure or F-score is the harmonic mean of precision and recall and it is also used to evaluate the performance of automatic keyphrase extraction. However, the F-measure of evaluation of keyphrase extraction is criticized (Torsten, 2009), as the F-measure evaluations misperforms in particular conditions. As the documents have varying numbers of keyphrase assigned by the manual annotators, a cutoff might distort results for some documents. consider a case where we always extract n keyphrases, but a document has less than n gold keyphrases assigned, then extracted keyphrases will always be wrong which affects the precision. So, for evaluating keyphrase extraction (Torsten, 2009), R-Precision is considered, where we extract the exact number of keyphrases for a document as in the gold standard. R-Precision can be defined as the precision at a cut-off R where R, is the exact number of phrases in the gold standard for a given document. Looking for the exact matches from the gold standard can under-perform the results of the keyphrase extraction system(Turney, 2000) as the extracted keyphrases can be longer or shorted or morphological

invariant of a phrase in the gold standard. This new approximate strategy is tested whether it is acceptable to human beings, with the morphological invariants got the highest agreement with 96% acceptance followed by longer phrases with 80% acceptance and shorter phrases with 44% acceptance (Torsten, 2009). The shorter phrases are left because they have less agreement. In this thesis, we are considering longer phrases and morphs for approximate matching. With morphological invariants, we are only considering a phrase as a keyphrase if its plural exist in the gold standard. A deeper investigation of morphological invariants is left for the future work.

5.1.5. Experimental Results

In an attempt to gain a better insight into the proposed keyphrase extraction systems, we are comparing the results with TextRank (Mihalcea and Tarau, 2004). As we wanted to ensure a fair comparison, the R-Precision is calculated for both TextRank for both exact matching and approximate matching. The R-Precision of exact $R - P_{ex}$ and approximate matching $R - P_{app}$ of our four approaches are reported. The biased approaches are reported with both weakly (W) and strongly (S) preferential.

- Phrase-frequency biased PhraseGraphs (PPGW and PPGS) as defined in section 4.2.4.1
- Term Frequency biased PhraseGraphs (TFPGW and TFPGS) as defined in section 4.2.4.2
- TFIDF biased PhraseGraphs (TFIDFPGW and TFIDFPFS) as defined in section 4.2.4.2

5.1.6. Discussion

Table 5.1 summarizes the results of our experiments.

We considered TextRank as a baseline in all of our experiments. The gold standard contains 4913 phrases extracted from the 500 INSPEC abstracts. Table 5.1 shows the number of phrases that match exactly, number of phrases that are longer than the phrases in the gold standard, number of phrases that are morphological invariants, R-Precision when exact

Table 5.1. Results1

	No. of Assigned Phrases	No. of Exactly Matched Phrases	No. of Matched Longer Phrases	No. of Matched Morph Phrases	$R-P_{ex}$	$R-P_{app}$	%ex-imp	%app-imp
Text Rank	4913	1341	445	2	0.2729	0.3635		
PG	4913	1369	535	2	0.2786	0.3879	2.08%	6.71%
PPGW	4913	1392	535	2	0.2833	0.3926	3.81%	8.01%
PPGS	4913	1394	535	2	0.2837	0.3930	3.95%	8.115%
TFPGW	4913	1425	593	2	0.2900	0.4111	6.26%	13.09%
TFPGS	4913	1426	593	2	0.2902*	0.4113*	6.33%	13.14%
TFIDFW	4913	1408	566	2	0.2865	0.4021	4.98%	10.61%
TFIDFS	4913	1409	566	2	0.2867	0.4024	5.05%	10.70%

matching is considered($R-P_{ex}$), R-Precision when approximate matching is considered($R-P_{app}$), % of improvement in R-Precision compared to our baseline which is TextRank. The last two columns in the table %ex-imp and %app-imp represents the relative change in $R-P_{ex}$ and $R-Papp$ compared to $R-P_{ex}$ and $R-P_{app}$ of TextRank respectively. For example, the %ex-imp of PG is calculated using the equation:

$$\%ex-imp = \frac{PG(R-P_{ex}) - TextRank(R-P_{ex})}{TextRank(R-P_{ex})} \tag{23}$$

where $PG(R-P_{ex})$ represents $R-P_{ex}$ with PG approach and $TextRank(R-P_{ex})$ represents $R-P_{ex}$ with TextRank approach. $R-P_{ex}$ improved at least by 2% compared to TextRank with the highest of 6.33% improvement. $R-P_{ex}$ slightly improved with PhraseGraphs compared to TextRank. The phrase-frequency biased PhraseGraphs improved performance slightly compared with the PhraseGraphs(PG). The TFIDF biased PhraseGraphs (TFIDFS) significantly

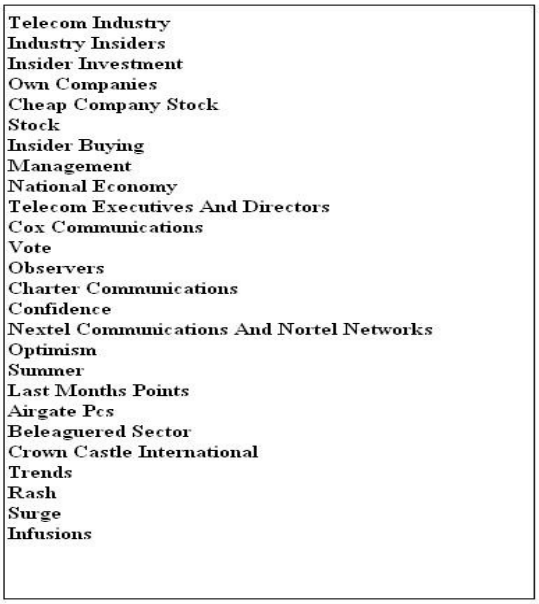

Figure 5.1. Ranked list of phrases using TFPGS for the abstract in figure 1.2

improved the $R - P_{ex}$ and $R - P_{app}$ by "5.05%" and "10.70%" respectively compared to TextRank. Term frequency biased PhraseGraphs (TFPGS) performed best with an $R - P_{ex}$ equal to 0.2902 and $R - P_{app}$ equal to 0.4113 showing an improvement of "6.33%" and 13.14% respectively. In figure 5.1, we showed the ranked list of phrases produced by term frequency biased PhraseGraphs(TFPGS) for the abstract in figure 1.2

In most of our cases, strongly preferential slightly performed better than weakly preferential. As the dataset used in this experiments are abstracts which are smaller in length and

Table 5.2. Results2

	No. of Assigned Phrases	No. of Exactly Matched Phrases	No. of Matched Longer Phrases	No. of Matched Morph Phrases	$R-P_{ex}$	$R-P_{app}$	%ex-imp	%app-imp
Text Rank	3865	1138	366	2	0.2944	0.3896		
PG	3865	1132	452	0	0.2928	0.4098	-0.54%	4.9%
PPGW	3865	1145	448	0	0.2962	0.4121	0.61%	5.77%
PPGS	3865	1144	448	0	0.2959	0.4119	0.50%	5.72%
TFPGW	3865	1195	516	0	0.3091	0.4426*	4.99%	13.60%*
TFPGS	3865	1196	515	0	0.3094*	0.4426*	5.09%*	13.60%*
TFIDFW	3865	1188	480	0	0.3073	0.4315	4.38%	10.75%
TFIDFS	3865	1190	481	0	0.3078	0.4323	4.55%	10.95%

usually discussing about a single topic, the probability of having dangling nodes is low. With longer documents, we strongly believe, a definite variation can be seen in the performance of weakly and strongly preferential approaches which is left for the future work. As explained in the section 1.1, authors of the INSPEC dataset also included keyphrases that are not in the document. In the dataset considered for the experiments, authors introduced 21.34% Phrases of gold standard which are not in the document. We removed the keyphrases which are not in the document from the gold standard and performed all of our experiments. The results are listed in table 5.2. As we considered R-Precision as an evaluation metric, there is no evidence that the performance of the system should improve. But, the results in table 5.2 showed a significant increase in performance compared to results in table 5.1. The improvement in performance shows the system has high precision metric too. Even, in this

case, both term frequency biased PhraseGraphs and TFIDF biased PhraseGraphs showed remarkable performance, which outperformed the TextRank by a wide margin. The term frequency biased PhraseGraphs performed best in all of our approaches. Though, the term frequency biased PhraseGraphs and TFIDF biased PhraseGraphs outperformed other existing approaches, a further investigation should be made to determine a better phrase weighting scheme, which is left for future work.

5.2. Chapter Summary

In this chapter, a novel keyphrase extraction approach has been introduced based on semantic relatedness and biased graphs. These approaches are introduced to find the phrases that strongly represent the document based on their semantic relatedness with other phrases in the document and their own frequencies. This approach helps in identifying the phrases that are both frequent and infrequent in a document. The infrequent terms are extracted if they have a strong semantic relatedness with other phrases and frequent phrases are extracted if they have higher frequency. We also compared the results with TextRank and showed the improvement in performance of our approaches.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, a comprehensive study of automatic keyphrase extraction using wikipedia-based semantic relatedness is explored. We expected the important phrases of a document have stronger relations with other phrases in the document. We also expected the important phrases of the document have higher frequency, and higher term frequency of individual terms in the phrase and higher TFIDF of individual terms in the phrase. Phrase Graphs has been built on individual documents and we expected the important phrases have stronger relation with other phrases. In brief, the new approach is built for key phrase extraction that gives importance to both frequent and infrequent phrases gets precedence in their own manner. We evaluated our results using a novel approach R-Precision and the results showed all our expectations are correct. The results showed significant improvement in the performance of our system. Thus, combining semantic resources like Wikipedia, using different variants of graph algorithms like pagerank based on the term frequencies resulted a better performing automatic keyphrase extraction system.

6.1. Future Work

In future, the proposed keyphrase extraction will be tested on Austin Papers, which helps in answering various historical research questions as explained in section 1.2. We want to introduce a better performing semantic relatedness metric, especially for phrases. For this purpose, we want to investigate with other existing semantic resources like Wikitionary, Wordnet and introduce a new framework which exploits all of these semantic resources. Also, a deeper insight into morphological variations will help to evaluate the real performance of our

proposed keyphrase extraction. Another research direction is to investigate how well the R-Precision is evaluating the automatic keyphrase extraction systems and how to improve the evaluation metrics for automatic keyphrase extraction if needed. One of the weaknesses of R-Precision is the lower discriminative power compared to Mean Average Precision(MAP). The major issue here is that MAP takes into account the ranking of phrases. So, in future, we want to investigate how MAP can be used as an evaluation metric for this task. A much important research direction is to see the performance of our system on longer documents, and how Weakly and Strongly Preferential approaches vary in results. We want to find a better phrase weighting scheme in future other than proposed Phrase Frequency biased, Term Frequency biased and TFIDF biased schemes. Finally, this thesis created a plenty of research directions to investigate in future for developing a better performing keyphrase extraction system taking advantage of our proposed system.

BIBLIOGRAPHY

[1] S. Banerjee and T. Pedersen, *Extended gloss overlaps as a measure of semantic relatedness*, Proc. of the 18th Int'l. Joint Conf. on Artificial Intelligence, 2003, pp. 805–810.

[2] Ken Barker and Nadia Cornacchia, *Using noun phrase heads to extract document keyphrases.*, Canadian Conference on AI (Howard J. Hamilton, ed.), Lecture Notes in Computer Science, vol. 1822, Springer, 2000, pp. 40–52.

[3] Paolo Boldi, Massimo Santini, and Sebastiano Vigna, *A deeper investigation of pagerank as a function of the damping factor.*, Web Information Retrieval and Linear Algebra Algorithms (Andreas Frommer, Michael W. Mahoney, and Daniel B. Szyld, eds.), Dagstuhl Seminar Proceedings, vol. 07071, Internationales Begegnungs- und Forschungszentrum fr Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[4] Franois Bourgeois and John-Claude Lassalle, *An extension of the munkres algorithm for the assignment problem to rectangular matrices.*, Commun. ACM 14 (1971), no. 12, 802–804.

[5] D.B. Bracewell, F. Ren, and S. Kuriowa, *Multilingual single document keyword extraction for information retrieval*, oct. 2005, pp. 517 – 522.

[6] Rudi L. Cilibrasi and Paul M.B. Vitanyi, *The google similarity distance*, IEEE Transactions on Knowledge and Data Engineering 19 (2007), no. 3, 370–383.

[7] Christiane Fellbaum, *Wordnet: An electronic lexical database*, Bradford Books, 1998.

[8] András Frank, *On kuhn's hungarian method - a tribute from hungary*, Tech. Report TR-2004-14, Egerváry Research Group, Budapest, 2004, `www.cs.elte.hu/egres`.

[9] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning, *Domain-specific keyphrase extraction*, IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (San Francisco, CA, USA), Morgan Kaufmann Publishers Inc., 1999, pp. 668–673.

[10] E. Gabrilovich and S. Markovitch, *Computing semantic relatedness using wikipedia-based explicit semantic analysis*, Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 6–12.

[11] Kazi Saidul Hasan and Vincent Ng, *Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art*, Proceedings of COLING 2010: Posters Volume, 2010, pp. 365–373.

[12] Katja Hofmann, Manos Tsagkias, Edgar Meij, and Maarten de Rijke, *The impact of document structure on keyphrase extraction.*, CIKM (David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, eds.), ACM, 2009, pp. 1725–1728.

[13] A. Hulth, *Improved automatic keyword extraction given more linguistic knowledge*, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (Japan), 2003.

[14] Anette Hulth, *Reducing false positives by expert combination in automatic keyword indexing*, Recent Advances in Natural Language Processing III, RANLP 2003, 2004, pp. 367–376.

[15] Glen Jeh and Jennifer Widom, *Scaling personalized web search*, WWW, 2003, pp. 271–279.

[16] J.J. Jiang and D.W. Conrath, *Semantic similarity based on corpus statistics and lexical taxonomy*, Proc. of the Int'l. Conf. on Research in Computational Linguistics, 1997, pp. 19–33.

[17] John Lafferty, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, Morgan Kaufmann, 2001, pp. 282–289.

[18] C. Leacock and M. Chodorow, *WordNet: An electronic lexical database - combining local context and wordnet similarity for word sense identification, in wordnet: An electronic lexical database*, ch. 11, pp. 265–283, MIT Press, 1998.

[19] Michael Lesk, *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*, Proc. of the 5th Annual Int'l. Conf. on Systems Documentation, 1986, pp. 24–26.

[20] Dekang Lin, *An information-theoretic definition of similarity*, Proc. of the 15th Int'l. Conf. on Machine Learning, 1998, pp. 296–304.

[21] Inderjeet Mani and Mark T. Maybury, *Advances in automatic text summarization*, 1998.

[22] R. Mihalcea and P. Tarau, *TextRank: Bringing order into texts*, Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing, July 2004.

[23] D. Milne, *Computing semantic relatedness using wikipedia link structure*, Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRSC) (Hamilton, New Zealand), 2007.

[24] D. Milne and I. H. Witten, *An open-source toolkit for mining wikipedia.in online proceedings of the new zealand computer science research student conference, auckland, new zealand.*, 2009.

[25] David Milne and Ian H. Witten, *An effective, low-cost measure of semantic relatedness obtained from wikipedia links*, 2008.

[26] Thuy Dung Nguyen and Min yen Kan, *Keyphrase extraction in scientific publications*, In Proc. of International Conference on Asian Digital Libraries (ICADL 07, Springer, 2007, pp. 317–326.

[27] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, *The pagerank citation ranking: Bringing order to the web*, Tech. report, Stanford University, 1999.

[28] Xuan-Hieu Phan, *"crfchuker: Crf english phrase chunker", http://crfchunker.sourceforge.net/*, 2008.

[29] ———, *"crftagger: Crf english pos tagger", http://crftagger.sourceforge.net/*, 2008.

[30] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner, *Development and application of a metric on semantic nets*, IEEE Transactions on Systems, Man and Cybernetics 19 (1989).

[31] Philip Resnik, *Using information content to evaluate semantic similarity in a taxonomy*, Proceedings of the 14th IJCAI (Montréal (Canada)) (C. Raymond Perrault, ed.), 1995, pp. 448–453.

[32] Michael Strube and Simone Paolo Ponzetto, *Wikirelate! computing semantic relatedness using wikipedia*, AAAI, AAAI Press, 2006.

[33] Zesch Torsten, *Study of semantic relatedness of words using collaboratively constructed semantic resources.*, 2010.

[34] P. Turney, *Extraction of keyphrases from text evaluation of four algorithms*, http://ai.iit.nrc.ca/II_public/extractor/reports/index.html (1997).

[35] Peter D. Turney, *Coherent keyphrase extraction via web mining*, CoRR cs.LG/0308033 (2003), informal publication.

[36] Xiaojun Wan and Jianguo Xiao, *Single document keyphrase extraction using neighborhood knowledge.*, AAAI (Dieter Fox and Carla P. Gomes, eds.), AAAI Press, 2008, pp. 855–860.

[37] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning, *Kea: Practical automatic keyphrase extraction*, CoRR cs.DL/9902007 (1999), informal publication.