Indian Statistical Institute
Kolkata

M.Tech. (Computer Science) Dissertation

Phrase Rank - An Iterative Graph-Based Algorithm for Unsupervised Key-Phrases Extraction

A dissertation submitted in partial fulfillment of the requirements
for the award of Master of Technology
in
Computer Science

Author:
Mayur Chhabra
Roll No: CS 1308

Supervisor:
Dr. Debapriyo Majumdar
Computer Vision and Pattern
Recognition Unit

CERTIFICATE

M.TECH(CS) DISSERTATION THESIS COMPLETION CERTIFICATE

Student : Mayur Chhabra (CS 1308)

Topic : Phrase Rank - An Iterative Graph-Based Algorithm for Unsupervised Key-Phrases Extraction

Supervisor : Dr. Debapriyo Majumdar

This is to certify that the thesis titled **Phrase Rank - An Iterative Graph-Based Algorithm for Unsupervised Key-Phrase Extraction** submitted by Mayur Chhabra in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work carried out by him under our supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in our opinion, has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other university for the award of any degree or diploma.

Debapriyo Majumdar

Date : 14^{th} *August*, 2015

Acknowledgement

First of all, I wish to express my sincere gratitudes to **Dr. Debapriyo Majumdar** for suggesting me this topic in the first place and helping me with the challenges that I have faced. Without his guidance and immense support, this project would not have been possible.

I am very grateful to all my classmates especially Chirag Gupta, Dnyaneshwar Patil, Subhadeep Ranjan Dev, Tanmoy Patra and Harmender Gahalawat for helping me in technical matters and discussing different ideas.

Abstract

The Key-Phrases are the set of important phrases from the text, where each phrase provide some unique and important information from the document and the complete set can represent whole document. The user can get a quick insight about the document by providing summary as the key-phrases. These Key-Phrases can be utlized for indexing and in other Information Retrieval Applications.

Many researchers suggested both the supervised and unsupervised techniques for extracting the key-phrases from the text document. Where as we focused our study only on the unsupervised algorithms. The state-of-art unsupervised algorithms suggested by researchers, are based on different models, some of them are based on simple clustering, while the others are designed on language models or graph-based models. We studied various graph-based algorithms that includes Text-Rank, Single-Rank and Expand-Rank algorithms and design a new enhanced graph-based model to extract the key-phrases from the text-document.

The old graph-based models used the undirected word-graph, that is able to capture only term-term association from the document. Our new enhanced graph-based model is able to represent more and better relationships between the terms by using directed wegithed graph. Apart from this our new model designed to constructs better multi-word phrases by proper validation checks and uses phrase-graph instead of word-graph. The purpose of using the phrase-graph is to capture the relationship between the phrases of text document, which is a better representation of the document than the word graph.

We evaluated our model by comparing the classification efficiency of generated key-phrases by our model with the generated phrases by old graph based models, and we also perform a manual evaluation to check whether our algorithm is able to generate better valid multi-word phrases. Both the evaluation results in the favour of our new model, We are getting 44.05% classification accuracy by our new model as compare to the 38.60% accuracy by the Text-Rank algorithm and 33.48% accuracy by Single-Rank algorithm, which is around 14% improvement over the Text-Rank algorithm and 24% improvement over Single-Rank algorithm. The manual evalution also shows that our new model is able to generate more valid phrases than the Text-Rank and Single-Rank algorithm.

Contents

Chapter 1

Introduction

1.1 Introduction to Key Phrases

The Key-Phrases are the set of important phrases from the text which can represent whole document. These Key-Phrases are able to summarize the whole text by providing brief context about the information present in the document.

Example 1.1-

Sample Document :

"The Indian Statistical Institute (ISI) Kolkata announced openings for the post of Junior Research Fellow and applications are invited from eligible candidates for recruitment to this post. The interested candidates may attend the walk-in interview "

Key-Phrases :

"Indian Statistical Institute", "Junior Research Fellow", "Recruitment", " Application","Walk-in interview".

Key - Represent something that have a great importance or can perform some unique great task.
Phrase - A single word or a small group of words standing together as a conceptual unit, typically forming a component of a clause.

Thus we can define **Key-Phrases** as the set of phrases where each phrase provides some unique and important context about the document, and the complete set is expected to provide summarize information about whole document.

1.2 Motivation

There are some texts from different domains which are usually labeled by the author such as Scientific papers, blogs, new articles. But many of the online contents are unlabeled, and It is always desirable to

provide the label to such contents, that can be used for indexing while searching or provide the summarize context about the document.

The summary in the form of Key-Phrases can give user a quick idea about the document so that user before going through the whole text can get some idea about what the document is all about. The online contents are huge so it is not possible to employed the human to do the manual labeling of the texts, Hence the need for automatically labeling these texts arises.

1.2.1 Applications

There are many natural language processing (NLP) and information retrieval (IR) tasks, which can be benefited by the key-phrases such as [3]:

1. Indexing and Searching :- If we knows the set of Key-Phrases of a particular document we can use these phrases to index our document for search, So we will save the time and space for indexing the document with whole content which also speed up the searching process.

2. Text summarization : :- As we already discussed that key-phrases are the set of phrases which can give a summarized contextual view of the complete document, and thus the Key-phrases can be used to defines the whole document.

3. Text categorization : :- Key-phrases also can be used to classify the document, as they supposed to be contain the all important information about the texts, hence they can be good feature set for documents categorization. Even in our experiments we categorized the documents by taking their key-phrases as the only feature set, the results shows increase in error rate, but still quite satisfiable.

4. Opinion mining : :- An another area where the Key-Phrases can be used instead of the whole document.

5. Faceted Searching : :- Key-phrases are the good candidates to be included in the facets, that can be used to make the browsing and navigation experience more efficient.

There are many more areas where these key-phrases can be used.

1.3 Problem Statement

"To design an Unsupervised Key-Phrases Extraction mechanism to automatically extracting the set of important key-phrases from the given Text Document. Our goal is to design such Key-Phrases extraction algorithms that should be independent of the text domain and should perform better then previous state-of-the-art algorithms."

Many researchers suggested both the supervised and unsupervised techniques for extracting the key-phrases from the text document. While supervised techniques always have some disadvantages as it requires a large set of data to be trained on, getting these large set of labeled datasets are difficult and the more complex task is to combine the texts from different domains ,as the supervised techniques always have biasedness towards the domain it is trained on. Thus the supervised techniques not appears as a good performer for this task of phrase-extraction for the general text documents, or some new domain content which it has not experienced at training time. The only alternative remains in this case is unsupervised models but those are not much trivial.

The unsupervised techniques on the other hand are completely based on some heuristics and assumptions which may be work well on some text domain but not others. There are many unsupervised techniques suggested by the researchers which includes graph-based ranking, language modeling and clustering based approach. All these techniques works effectively for only some particular text domains and document size.

In this paper we suggested one Iterative graph-based model which is a variants of text-rank model, that is one of the state-of-the-art approach for extracting the key-phrases from the plain text documents. The text-rank algorithm is based on the heuristic that the most important words of the texts are those which frequently occurred in the document and also surrounded with the more unique words. The assumption is completely non-intuitive but experiments shows very good results by using this algorithm.

Our models are different from the text-rank algorithm on basically on two aspects. One is that our model provides the directed link between the nodes of graph, where the text-rank only uses undirected edges, and the second difference is that our Term-Graph also included multi-word phrases, thus we called our graph a phrase-graph instead of word-graph used in Text-Rank Algoritm.

We have evaluated our models by classifying the text document based on the extracted key-phrases by Text-rank, Single rank and our new suggested model, and found that our model perform slightly better in classifying the documents and gives less error rate. We have also performed a manual evaluation to evaluate that which model generates the more valid and meaningful multi-word phrases, and we got the results in favour of our model.

Chapter 2

Related Work

There are many unsupervised algorithms suggested by the Researchers for extracting the Key-Phrases from the text documents, that are based on different models. some of them are based on simple clustering, while the others are designed on language models or graph-based models. We are mentioning here some of the fundamental models to understand the basic foundation of unsupervised approaches to extracting the Key-phrases, Page-Rank algorithm on which many graph-based algorithms for Key-Phrase extraction are based on, following with some state-of-art algorithms for extracting the key-phrases [3] [4].

2.1 Tf-Idf

Tf-Idf[10] is the most basic and fundamental approach for extracting the important phrases from the text. The algorithm based on the phenomenon that if a phrase is significantly important for a particular text, then it should be repeatedly occurred many times in the document. It is the most crude approach but empirically works very well even better than many state-of-art models with some text domains.

Technically the Tf-Idf model works by counting the frequency of Term (where each term is a candidate word) denotes as Tf and Inverse Document frequency denoted as Idf and then calculate TfIdf score for each term as

$$TfIdf_t = Tf_t \times log(D/D_t) \tag{2.1}$$

Where D is the number of documents and D_t is the number of documents which contains the term t. After computing the TfIdf score for each term t in document D, it computes the score for each candidate phrase by summing the score of its constituents words and thus return the top-k scorers phrases.

This approach is the basis for many other Key-phrase extraction models. The term frequency is the most fundamental feature that almost all Key-phrase extraction models considered directly or indirectly.

2.2 Page-Rank

The Page-Rank[1] algorithm is used to rank the pages over the web. According to Google: "PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites."[Wikipedia].

The algorithm uses a graph-based model and follows these steps to compute the rank of pages.

1. for each page P_i there exists a corresponding node N_i in the graph
2. there is an edge E_{ij} if there exist a link from N_i to N_j, which may be weighted or unweighted.
3. Set the initial rank of each page = 1/N
4. The Rank for each page is thus calculated in each iteration by a famous random surfer model.

$$PR(P_i) = \frac{1-d}{N} + d \sum_{P_j \epsilon M(P_i)} \frac{PR(P_j)}{L(P_j)} \tag{2.2}$$

   Where i = 1,2,3,...,N, N is the total number of pages, $M(P_i)$ is the set of all pages which have links to P_i, $L(P_j)$ is the number of links to all the other pages from P_j, and d is the residual probability. [Wikipedia]

5. Run the step 4 for till there is no change between $PR(P_i)$s or for some fixed iterations ie. 20 or 30 times.

2.3 Text-Rank

The Text-Rank[8] approach is designed on graph based model, that is derived from the well known Page-Rank algorithm used to rank the interlinked pages on the web. Where the Tf-Idf approach only takes term frequency into account the Text-Rank approach also consider that the term should co-occur with many other unique terms. How can this feature can help..??

Assumes, that if a particular term occurs many times in the document but all the time it is surrounded by almost same set of other terms, then the term may not be much important, as all the time the term is likely to be repeated in same context. But if the term surrounded by almost unique terms each time, then It may have some significant importance as it occurred many times, in addition each time in different context or we can say each time we are talking about something different but using the same term again and again so it may have some importance in the text.

The Text-Rank tries to captures the term-term(word-word) association from the text, by representing the text-document into Word-Graph and then find the most important words analogous to the most influencial pages in case of PageRank.

Word Graph : The Word-graph of a Text Document D can be defined as graph G(V,E), where there is a corresponding node in the graph G for each unique word in document D, and There will be an undirected edge between the two nodes if their corresponding words co-occured with in a particular window size W.

Example 2.1. Sample Document:

"Social media is a place where users present themselves to the world, revealing personal details and insights into their lives. Personality has been shown to be relevant to many types of interactions; it has been shown to be useful in predicting job satisfaction, professional and romantic relationship success. Until now, to accurately gauge users personalities, they needed to take a personality test. This made it impractical to use personality analysis in many social media domains. We present a method by which a user's personality can be accurately predicted through the publicly available information on their social media profile."

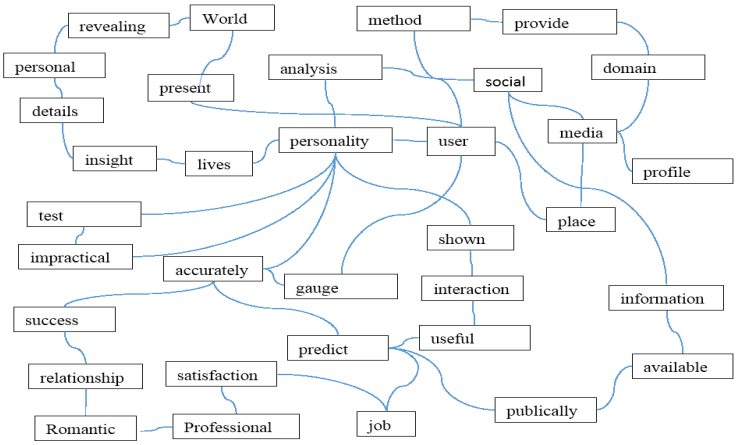

Figure 2.1 : Word-Graph for Sample Document with window wize 1

Method for Text-Rank :

1. Lexical unit selection , Select the Set of candidate words that can be used to contruct the key-phrases.

2. Construct a word-graph from candidate words from the Document and then compute the score for each candidate word by

$$S(V_i) = (1-d) + d * \sum_{j \epsilon In(V_i)} \frac{1}{Out(V_j)} * S(V_j) \tag{2.3}$$

Where $S(V_i)$ is the Score for vertex V_i corresponding to term T_i. d is the dumping factor that can be set between 0 and 1. $Out(V_j)$ is Set of all the vertex m, that have an edge from V_j to V_m and $In(V_j)$ is Set of all the vertex n that have an edge from V_j to V_n.

3. And finally all the phrases which only contains top k scored words are returned as the set of key-phrases.

2.4 Single-Rank

The Single-Rank[12] algorithm is similar to the Text-Rank algorithm but with some modifications
1) **Edge Weights :** In Text-Rank all the edges have equal weights but in Single-Rank edges have weight equal to the number of times the corresponding words co-occur within the provided window range.
2) **Phrase Construction and selection :** In Text-Rank only Top ranked words participate in creating Key-phrases while in the Single-Rank first all the words participate in creating phrases and then the phrases are scored as :

Score(Phrase P) = Score(P1) + Score(P2) + ... + Score(Pm), where m is the size of phrase P and Pi= ith word in phrase P, for i=1...m

Example : Score("social media profile") = Score("social") + Score("media") + Score("profile")

after scoring of the phrases, all the phrases are ranked according to their score and k top-ranked phrases chooses as the Key-Phrases.

2.5 Expand-Rank

ExpandRank [12], is also a variant of text rank with some extended feature. The algorithm first find the K-nearest documents of the test document and then it combines those K+1 documents and make a graph from them similar to text-rank, but the weight of the edge is computed as

$$w(v_i, v_j) = \sum_{d_k \epsilon D} sim(d_0, d_k) * freq_{d_k(v_i, v_j)} \tag{2.4}$$

where w(vi, vj) is the edge weight between Vertex i and j, and sim(d0 d1) is a similarity measure that is used to compute the top-K similar document for the test document, and $freq_{dk(vi,vj)}$ denotes the frequency of vi and vj co-occurring in the document dk. The Extended rank works better than normal text-rank algorithm with almost every dataset, but we need to have a big set of corpus of the files so that we can get best-k matches for the test document.

2.6 Clustering based approach

The Clustering based approach [6] first cluster the document terms on the basis of their co-occurrence in a particular window size w, or using Wikipedia based statistics of semantically related word. Before that we can preprocess the text to exclude some drop-words from the document, after having the different clusters, It choose one term from each cluster. as each cluster have, and Finally it returns all the candidate phrases which have one or more exemplar term.

Chapter 3

Our approach: Phrase-Rank

We planned to focused our study on the Graph-Based models only as many of the state-of-art algortihms for key phrase extraction are designed over it and these models also performs well. [3]

We found that all the graph-based algorithms are in general comprises of two sub-tasks

1. Computing the score for candidate words in the document.
2. Construct the phrases, rank them and found top k-phrases.

We disigned a new Graph-based model for Key-Phrase extraction, which tries to enhance the performance of each of these tasks by applying some improvements over previous graph-based models. Our new model is designed over Text-Rank model which is the most basic Graph-Based model, over which other Graph-based models are designed. Our new model is capable of :

3.1 Assigning right direction and weights to the nodes of the graph representing the text

In previous graph-based algorithm the word-graph only represent the word-word association in text, where all the words occurs together within a particular window size have an undirected edge with equal weights.

The goal of our new model is to construct the term-graph, which is able to represent the document structure better by capturing more semantic and syntactic relationships between the terms in document.

In PageRank, an inlink means a page is more important, so it gets more PageRank. In the word-graph scenario, an inlink should correspond to a more important word. Typically in sentences, nouns, adjectives and verbs should have different weights. While any pre-defined set of weights would not be able to perfectly simulate the right weights for the words of any sentence, we try to find reasonable uniform weights for the words based on their PoS tags. We have experimented different weights on each direction

of the links between the nodes and realized that provided more weight to the inlinks to verb and noun, while less weight assigned to inlinks to adjective give better performance, provided more weightage to the inlink to noun than verb even give more better results.

3.2 Designing a systematic approach for constructing phrases and ranking them

The other problem we found in previous graph based algorithms that the construction of the phrases from the top scored key-words is completely ad-hoc. Where the consecutive candidate words are combined and treated as single phrase. Additionally The phrase-selection part is also have some flaws.

In Text-Rank algorithm all the consecutive top scored words are combined together and printed as key-phrases. There is no validation check for constructed phrase structure and no seperate scoring and ranking for phrase. So even a phrase which occurs only once in the docuement, and thus not a good candidate to become a key-phrase, still come up as a key-phrase because its constituents words occured frequently in text.

For Example : "Clustering Algorithm", "Data Mining" are the two phrases in the document.

1. The Phrase "Data Mining" appears 3 times in document, and co-occured with 8 unique words.
2. The phrase "Clustering Algorithm" exists only once and co-occured with 2 unique words.

but words "Clustering" and "Algorithm" independently exists many more times than words "Data" and "Mining", hence their score is high. Aaccording to the concept of graph-based ranking the phrase "Data Mining" should get more importance but as constituents words of "Clustering Algorithm" have more score, Thus phrase "Clustering Algorithm" got more importance than phrase "Data Mining".

The Single Rank also faced the same problem in phrase selection additionally the single rank always give more priority to the longer phrases, which is not a good selection precedure.

Our new model first construct the phrases with high scored words and some intial validation check. First it Provides intial score to the phrases, and then the phrases pases to the further levels where they all are treated as single terms. Then we construct the graph with these terms, and we called this graph a **Phrase-Graph**, as each term in this graph is a complete phrase.

3.3 Capture Relationship between phrases

With the help of Phrase-Graph we tried to capture relationships between the phrases instead of just capturing the relationship between words only. So now the graph-based scoring and ranking algorithm can now directly applied on the phrases, and we may get more justified ranking for multi -word phrases.

Where each multiword-phrase is treated as a single term unit, and there is not favouritism on the basis of their lenght or constitutents words individual scores.

We designed two different graph based models for Key-Phrase extraction, applying the above discussed features.

3.4 Model : Phrase-Rank

Our model extract the Key-phrases in three phases.

- Phase 1 : Start with the directed and weighted word graph, to get some initial score for words.
  Construct multi-word phrases with some validation checks and provide them some initial score.
- Phase 2 : Incrementally move from word graph to phrase graph.
- Phase 3 : Compute score for phrases and rank them.
  Return Top k phrases as Key-Phrases.

3.5 Method 1 : Additive Method

In this method, first we are starting with a term-graph, where each term is corresponding to a candidate word in text and then we are extracting the top phrases by the text-rank algorithm. After that in each iteration we are adding one phrase to the term-graph, if the phrase after adding to the term-graph get high score by random-surfer algorithm, then we will keep that phrase in the term-graph for the next iteration, otherwise we discard it.

3.5.1 Algorithm

Algorithm 1 Phrase-Rank 1

1 Select lexical units (eliminate stop-words, allow words with particular POS-Tags).

2 Let G be a Word-Graph, with all the candidate words.

3 Run the Random-Surfer algorithm on graph G, get the score for each word and construct the set of multi-words phrases P_C with only top-scored key-words and some basic validation checks.

4 Assigns some initial score to the multi-word phrases. (with lengthwise multiplier)

5 For each candidate phrase p_i in P_C, in descending order of their score

5.1. Construct Term-Graph G_{pi}, where all terms are candidate words, but phrase p_i is considered as single term.

5.2. Compute score of each term by applying random surfer algorithm on G_{pi}. If score of pi is high, then include pi in graph G_{pj}, j>i. Otherwise discard p_i.

6 Let P_k is Set of top k phrases, computed by the Page-Rank algorithm on the graph Gpn, Return P_k as set of key-phrases.

3.5.2 Details

Step 1 : we have taken only the Noun, Verbs and adjectives as the candidate words, and created the word graph G.

Step 3 : first we extract the high scored words(The number of words selected here should be atleast 3-4 times of the targated number of phrases) from text using random surfer algortihm on graph G, Then we form bigrams, trigrams and quadgrams from the text by using only high scored words, that pass some initial criteria.

1. All Bigrams must only contains high scored words.
2. All Trigrams must contains at least Two high scored words, and can contain a non-candidate word only in between other two words.
3. All Quadgrams must contains at least Three high scored words, and can contains other non-candidate words only either at 2nd or 3rd place.
4. All the Bigrams, Trigrams and Quadgrams should not contain the adjectives in last position.

Step 4: The initial score to the phrases assigned here, is equal to the average weight of all the candidate words it has.

The multiplier provided to the phrase score depending on their word-length is to resolve the conflict in case of nested overlapping phrases, which have same score, by choosing the longer phrase.

Example :

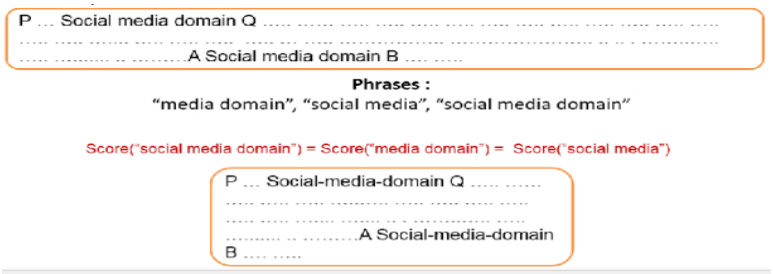

Figure 3.1 : Resolving overlapping conflict between nested phrases.

Step 5 : The Phrases including in the Phrase-Graph in decreasing order of their score to resolves conflict between overlapping phrases, by giving priority to high scored phrase among them.

Example :

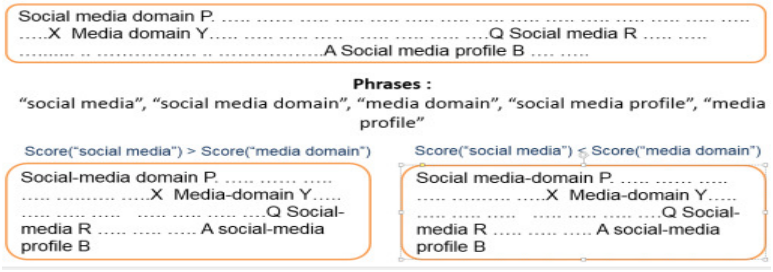

Figure 3.2 : Resolving overlapping conflict between phrases

The figure 3.2 and 3.3 shows overlapping between phrase “Social media domain”, “Social media”, and ”media domain” and how it is handled by our algorithm. our algorithm choose the phrase which have more score and ignore the other overlapping phrase.

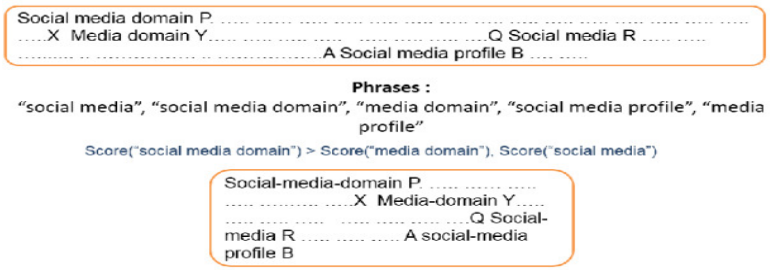

Figure 3.3 : Resolving overlapping conflict between phrases

3.6 Method 2 : Survival / Elimination Method

Here we are also starting with a word-graph, but in each iteration we only temporarily add a phrase P_i to the word-graph, and if that phrase P_i got good rank then we keep it in the Candidate Phrase Set P, otherwise we eliminate it. So at last we created a phrase-graph PG with all the survived phrases in set P and other words, and extract top ranked phrases by applying random surfer algorithm on graph PG, and return them as set of key-phrases.

3.6.1 Algorithm

Algorithm 2 Phrase-Rank 2

1 Select lexical units. (eliminate stop-words, allow words with particular POS-Tags)

2 Let G be a Word-Graph, with all the candidate words.

3 Run the Random-Surfer algorithm on graph G, get the score for each word and construct the set of multi-words phrases P_C with only top-scored key-words and some basic validation checks.

4 Create an empty set P.

5 For each multi-words phrase p_i in P_C (Allowing more selection ratio)

5.1 Construct Term-Graph G_{pi}, where all terms are candidate words, but phrase p_i is consider as single term.

5.2 Compute score of each term by applying Page-Rank on G_{pi}. If score of p_i is high, then put p_i in set P with their score.(with lengthwise multiplier)

6 Construct Phrase-Graph G_P, with all the multi-words phrases in P and other independent words. Phrases included in the graph in decreasing order of their score.

7 Compute the score of all the terms in graph G_P, and fetch the top K terms as set of key-phrases, let the set known as Pk.

8 Finally form graph G_{P*}, the Phrase-Graph with all the multi-words phrases in Pk and other independent words.

9 Let P_{k*} is Set of top k phrases, computed by the Page-Rank algorithm on the graph G_{P*}, Return P_{k*} as set of key-phrases.

3.6.2 Details

Step 1-3 are the same as Method 1.

Step 5 : Allowing more selection ratio is necessary here and should be atleast 3-4 times to the targated number of phrases needs to be extracted from text. This is because in Word-Graph where we have only one multi-word phrase and all the other terms are words, the score of the words will be very high as compare to multiword phrase, so it will be very difficult for a phrase to come in top-rank even when the phrase is very important, hence we should allow more ratio band for the multi-word phrases to be selected in this step.

Example : If our target is to select top 10% phrases of the total terms, then we should select a multi-word phrase in this step ,even it able to get rank in top 30% or 40%.

Step 8 : Here we are actually breaking those multi-word candidate phrases in words, which are not present in set of key-phrases extracted in step-7 and construct the graph once again, So to give their constituents words now one last chance to compete again to become a key-phrase, example 3.4 shows why this step is important. We have found by our evaluation that after this step our results got better by a big factor.

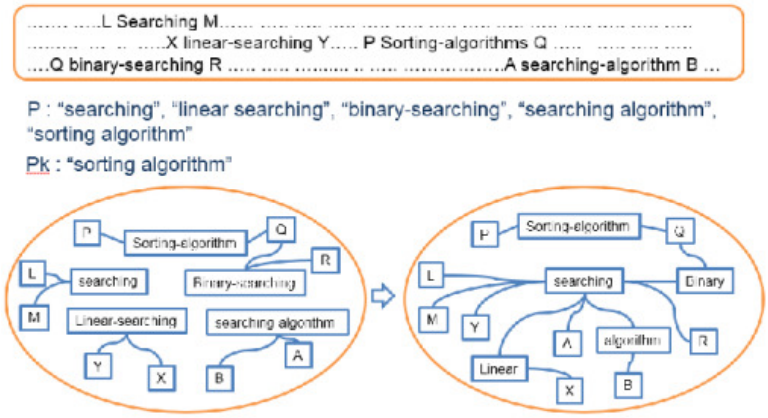

Figure 3.4 : Breaking multi-word phrases that are not in set of key-phrase , again into words and generate phrase-graph

In figure 3.4 we can observe that after breaking the non key phrases into words again, the linking of the searching increases and so it should get high score now and thus become a good candidate to become a key-phrase.

Chapter 4

Experimental Evaluation

4.1 Text Rank With Weighted and directed edges

The Key-Phrase extraction by Text-Rank algorithm comprises of two subtask as already discussed. The one is to extract the key-words from documents and the other task is to construct the key-phrases from those key-words.

We have seen the phrase construction part is very ad-hoc in Text-Rank, which is only dependent only on the key-words selected and there is not any further scoring and ranking for the phrases once they are created. Thus the only main part of the algorithm is to select good key-words.

We have experimented different direction and weights specification to form the word-graph from text Document. So compare the performance of any such two specification by comparing that which specification produce better key-words. We can say a particular set of words A is better than some other Set-B, if Set-A is providing better contextual information about the text than the Set-B. Thus from our belief a better Set-A should be a better feature set to classify the document than the Set-B. Hence if claffication accuracy of phrases generated by Text-Rank algorithm with some specification S_1 is better than the S_2 then S_1 is considered as better specification.

By assuming this behavior we evaluate our experiments on text-rank with various weights and directions specification by comparing their ability to generate key-words that can classify the document set with better accuracy.

4.1.1 Classification tool used

To perform the classification task we used **Weka** [1], which is a famous open source machine learning software issued under the GNU General Public License that have implemented various machine learning algorithms. **Weka** provides lots of algorithms to classify the text data with various customised

[1] www.cs.waikato.ac.nz/ml/weka

and preprocessing options. To classify the text document we first need to break the text document into word vector on the basis of some specification, before giving those as the input document to the classifier.

We choose the **Naive Bayes classifier** algorithm from **Weka** to perform the classification of our documents by using only key-phrases from the document, and evaluate the results on the basis of overall accuracy rate of classification of documents in complete data set. The reason of choosing the Naive Bayes is that it is a generative classifier which is also well suited for the text-document. Generative model is good choice for evaluating the key-phrases as we expects the same type of behaviour from our key-phrases that they should be used as a good model to generate the original text.

4.1.2 Data Set Used

20-newsgroups Data Set[2] **:** The 20-newsgroups Data Set is a well known text data set for experimenting the machine learning experiments like clustering and classification which is with the best knowledge is collected by the Ken Lang. The dataset comprises of 20,000 documents that contains news articles as reflected by the name itself and are evenly distributed among 20 different newsgroups. The data set is freely available.

4.1.3 Specifications and Results

We experiments on various weights and directions specification on word-graph, all these specification we have taken on the basis of PoS-tag only.

Some of the specification are shown below, and the classification accuracy of their corresponding generated key-phrases is shown in Table 1. Here we are taking only noun, adjectives and rare verbs (dropping some common verbs from the text) as the candidate words that compete to become a key-phrase.

To evaluate the different specification model, We have extracted key-words from each document of the data-set for both test and training data. Where the number of key-words are 20% of the total candidate words in document.

4.1.3.1 Specifications

Specification 1 'Edges between all the words have equal weights and the edges are undirected. (Text-Rank Algorithm)

Specification 2-5 Edges are directed, and their weight are depends on their POS tags.

[2] www.qwone.com/~jason/20Newsgroups/

Weight(a->b, where b is ...)	Spec 2	Spec 3	Spec 4	Spec 5
noun	1.0	1.0	0.6	0.6
verb	1.0	0.6	0.4	1.0
adjective	0.2	0.2	0.2	0.2

Specification 6 Edges are directed, and their weight are depends on their POS tags. Weight($a->b$, where a and b are ...)

a / b	Noun	Verb	Adjective
Noun	1	0.5	0.3
Verb	1	1	0.2
Adjective	1	0.5	0.8

4.1.3.2 Results

Spec 1	Spec 2	Spec 3	Spec 4	Spec4	Spec 5
47.57	49.52	50.24	49.92	46.89	49.96

The above results clearly shows that the key-phrases generated by some weight and directions specification on word-graph with Text-Rank give better results for classification than the simple Text-Rank algorithm which uses undirected and unweighted graph.

We can also observe that the edges towards noun given the higher weightage than verbs performs well in classification, while in the other results when edges towards adjectives were given higher weightage the classification accuracy went even worst.

So our first experiment was successful in showing that by using directions and weights with the word-graph in the Text-Rank can give us the better results for Key-Phrases Extraction, if we consider classification results as a measure of its goodness. And shows that with directions and weights we can represent a better semantic structure of a graph than simple undirected graph which represent only word-word association between them.

4.2 Our Method : Phrase Rank

We implemented both of our methods Method 1 and Method 2, That uses the phrase-graph, with the same weights and directions specifications, and evaluate them against the previous methods of key-phrase extraction. As the sole purpose of using the phrase-graph in our algorithm is extracting the more meaningful phrases by constructing the multi-word phrases in better way, instead of just combining the consecutive key-words to get the phrases.

By considering the main goal of using phrase-graph that is constructing better multi-word keyphrases, we decided to evaluate our results on two methods.

1. Classification Results : We classify the documents by taking only the key-phrases instead of whole document for both testing and training set. The Data-set and classification tool we used is the same that we used in above result. The only difference is that in the previous classification the features are key-words while for this classification we taking the key-phrases as the features.

2. Manual Evaluation of the multi-word key-phrases : We manually evaluated the multi-word key-phrases generated by each algorithm and specification, and compare to get that which algorithm generate more meaningful multi-word phrases.

4.2.1 Configurations

Ratio for Phrase Extraction

- **Initial Ratio** : 0.3, or 150 phrases whichever is lesser. for stage 1
- **Filter Ratio** : 0.3, or 150 phrases whichever is lesser, for stage 2 (used only in Algorithm 1)
- **Final Ratio** : 0.1 or 50 phrases whichever is lesser, for stage 3

Multiplier for multi-word phrases to give them higher privilege to be included in the graph before others.

- Trigrams gets a multiplier of 1.1 and
- Quadgrams gets a multiplier of 1.2

We decided to have key-phrases in the proportion of Final Ratio i.e. 10% of candidate phrases and max 50 in count. In previous stages we kept the extraction ratio higher, the reason behind is that our target number of final phrases are 10% but if we select the 10% high ranked keywords in the initial stages, and then when we combine them to make multi-word keyphrases, we will get less number of multi-word phase than 10% ratio of total candidate words. And we want that we should have around 10% most promising multi-word phrases in our list of candidate phrase in the final step, So the final competition will be between the top 10% multi-word phrases and all other single word phrases.

4.2.2 Results

4.2.2.1 Classification Accuracy

We computed the classifcation accuracy of the data-set by considering only the key-phrases generated by the Algorithm 1 and Algorithm 2 with all the **same specifications used in above modified Text-rank algorithm with one extra specification** which is generally used by Single-Rank algorithm.

Specification 7: The weights of the edge between two term nodes of the graph is equal to the number of times they occurred together in a particular window size. Here we are also comparing our results with the results of Single-Rank, because here our main goal is to compare the phrase-construction efficiency of the algorithm, and the phrase construction and extraction part of the Single-Rank algorithm is different from the Text-Rank algorithm. From the above classification results where the key-phrases generated

Table 4.1: Classification Accuracy

	Spec 1	Spec 2	Spec 3	Spec 4	Spec 5	Spec 6	Spec 7
Text-Rank	38.61	40.33	42.06	41.81	38.82	42.00	38.10
Single Rank	33.49	36.91	33.61	36.13	36.00	35.80	33.22
Algorithm 1	41.10	42.86	43.92	43.58	41.45	43.60	41.36
Algorithm 2	41.93	43.41	44.06	43.79	41.74	43.85	41.77

by different algorithms are only taken as the feature set, we can conclude that the our algorithms which uses the phrase graph able to provide better results from Weighted Text-Rank with each specifications.

The best classification result we get from Phrase-Rank is 44.05% as compare to the general Text-Rank algorithm which gives only 38.6086% classification accuracy, hence we are getting around 14.1% improvement in classification result.

classification results for the key-phrases generated by the single-rank algorithm is even worse, because the single rank algorithm the score of the phrases are very much biased towards the longer phrases. Thus the smaller phrases which are really good candidate and deserve to be a key-phrase dominated by the longer phrase that even have less important words.

4.2.2.2 Manual Evaluation

The other factor we have considered beside of classification accuracy, The multi-word key phrases generated by the algorithm should be valid, proper constructed, make some sense independently and should not be too general, Then only it we can say it a good key-phrase.

So we prepare a set of multi-word Key phrases generated by all the above algorithms with different specification given and manually evaluate them to check whether the phrases are properly constructed and meaningful.

To prepare the evaluation Set we taken 40 documents from 10 different categories of the 20-newsgroup dataset, and generated the key-phrases from the discussed algorithms. From all the algorithms and specifications by taking 10% phrases of the total candidate words in the text we could able to generate 250 phrases at least. So we took 250 phrases from each of these generated key-phrases and union them to make a single set of phrases. The final evaluation set contains 1517 multi-word phrases.

We get evaluated our evaluation set by 5 evaluators, where each single evaluator assigned a binary

rating to the phrases i.e. 1 if valid o/w 0, so for each phrase have a score from 0-5. We have taken average score of phrases from each of the set of key-phrases and shows the comparative results in Table 4.2 .

	Word Graph		PhraseGraph-1		PhraseGraph-2	
	Avg Score	Valid %	Avg Score	Valid %	Avg Score	Valid %
Spec 0	3.47	48.4	3.78	55.6	3.81	58.4
Spec 1	3.35	44.8	3.68	55.8	3.79	55.6
Spec 2	3.43	47.6	3.69	55.6	3.74	56.4
Spec 3	3.37	47.6	3.76	57.2	3.81	56.8
Spec 4	2.91	31.2	3.80	43.2	3.35	41.6
Spec 5	3.56	50	3.80	57.6	3.76	55.6
Single Rank	2.94	31.6	-	-	-	-

Table 4.2: Result of Manual Evaluation

We are considering a phrase is valid if it have score 4 or 5, means atleast 4 out of 5 people mark that phrase as valid.

With word-graph the best valid % we are getting is 50, which means that half of the generated phrase are valid, that is also with only one specification, with all other even more than half phrases are not valid. The best score we are getting by Phrase-Rank Algorithm1 is 57.6 and Algorithm2 is 58.4, shows 15.2% and 16.8% improvement over Text-Rank respectively.

From the above results of manual evaluation we can see that in almost all the cases our Phrase-Rank algorithms generates more meaningful phrases as compare to the Text-Rank and Single-Rank algorithm which uses word-graph.

Chapter 5

Conclusion

We have developed Phrase-Rank algorithms for extracting the Key-Phrases from the Text Document, which is based on graph-based model and inspired from the Page-Rank,Text-Rank and Single-Rank algorithms that are some state-of-the art algorithms for key-phrase extraction.

The purpose of our new apporach of Phrase-Rank was to extract more better key-phrases from the text document, that contains more contextual information about the document and also more valid and meaningful phrases. We examined our model on each of these aspects by performing classfication and manual evaluation and found our model performing better than the previous graph-based models.

By generated Key-phrases from our Phrase-Rank model we are able to get around 14% improvement in classifcation accuracy over the Text-Rank algorithm and 24% improvement in accuracy over Single-Rank algorithm. The manual evaluation also shows that our model is able to generate more valid phrases than the previous graph based models.

Finally we can conclude that capturing better structure of the document by using directed edges and providing appropriate weightage to the each link between the nodes of graph can leads to better key-phrases extraction. We have also successfuly concluded that the phrase-graph which represent the relationship between the phrases directly, provide us more better structure of the document and thus give better, unbiased and justified scoring and ranking for all multi-word phrases as well as single words than the old graph-based algorithms that uses word-graph.

Bibliography

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks, 30(17)*, pages 107–117, 1998.

[2] Gordon W. Paynter Ian H. Witten Carl Gutwin Frank, Eibe and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. pages 668–673, 1999.

[3] Kazi Saidul Hasan and Vincent Ng. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 365–373. Association for Computational Linguistics, 2010.

[4] Karen Spärck Jones. Automatic summarising: The state of the art. *Information Processing & Management*, 43(6):1449–1481, 2007.

[5] Niraj Kumar and Kannan Srinathan. Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 199–208. ACM, 2008.

[6] Peng Li Yabin Zheng Liu, Zhiyuan and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Lan- guage Processing*, pages 257–266, 2009.

[7] Yutaka Matsuo and Mitsuru Ishizuka. Key-word extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, pages 13(1):157–169, 2004.

[8] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[9] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326. Springer, 2007.

[10] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[11] Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. *In Proceedings of the ACL Workshop on Multiword Expressions.*, 2003.

[12] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860, 2008.

[13] Wikipedia. Automatic summarization. https://en.wikipedia.org/wiki/Automatic_summarization.

[14] Dominique Fontaine You, Wei and Jean-Paul Barth‘es. Automatic keyphrase extraction with a re-fined candidate set. pages 576–579, 2009.