Automatically identifying short key-phrases that characterise long texts

Grecia Madeleine Vazquez-Sanchez

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh

2015

Abstract

In this work, we present three methods for key-phrase extraction (KE) in research papers. Several methods for automatically extracting important key-phrases of documents have been already developed; however, scientific papers represent a special challenge because their key-terms have a degree of uniqueness that distinguishes the particular topic of the paper from many others. In this work, our aim is to develop a method to extract key-phrases that characterise a research paper; this means, we desire to capture the key-phrases that are highly representative of the current paper, but not representative of many other papers. We detail three approaches: (1) The use of centrality measures, specifically betweenness to select the important nodes from a graph of candidate key-phrases. (2) We propose a novel unsupervised method for KE that has its foundations on betweenness centrality measure, but with a slight modification to measure how difficult is to pass from one document to another through a key-phrase. With this method we intend to capture the uniqueness of a key-phrase. (3) We implement the weighted sum to combine the features of TF-IDF (our baseline) and betweenness centrality in order to give a synthetic score for each key-phrase. This score is used to re-rank all the candidate key-phrases.

In the body of this work, we describe the theory behind graph-based methods and we detail the conceptual design of our methods. We evaluate each of the three approaches and we compare the results with well-know state-of-the-art methods for key-phrase extraction. The *contrastive betweenness centrality*, our new method, overall has a better performance than other centrality measures, but is not able to outperform the TF-IDF baseline. Weighted sum outperforms TF-IDF.

Acknowledgements

Firstly and foremost, I would like to express my gratitude to my supervisor, Dr. Charles Sutton. His continuous support, and his patient and careful guidance have been fundamental to shape this work. I would also like to express my constant appreciation to the faculty of the School of Engineering of the UNAM, mainly Dr. Jesus Manuel Dorador-Gonzalez, for providing always valuable advice and support for my professional life.

Finally, my deepest gratitude is to my beloved family and my friends in Mexico, especially Dad and Mom, for their unconditional support, prayers and encouragement during this and every stage of my life.

This work is supported by the Consejo Nacional de Ciencia y Tecnologia (CONACyT) grant number 610053, and by the Secretaria de Educacion Publica (SEP) through the programme Beca Complemento 2014 grant number BC-2557. This material is based upon a postgraduate programme supported in part by the University of Edinburgh through the Edinburgh Global Latin-American Master's Scholarship.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Grecia Madeleine Vazquez-Sanchez*)

Table of Contents

Chapter 1

Introduction

A key-phrase is a sequence of words that describes the content of a document. We call key-phrase extraction (KE) to the task of identifying automatically these descriptive terms from the body of a document. Key-phrases provide compact, still rich information of the text, for this reason, KE has extensive applications in natural language processing (NLP) problems such as text summarisation, document indexing, document clustering, topic detection, document retrieval, query expansion, text mining, content-based tag recommendation, among many others.

Due to the growing amount of electronic documents which usually does not include a set of key-phrases, manual assignment of key-phrases is not only costly, but inefficient. Therefore, the research community has developed numerous methods for automatic key-phrase extraction. Supervised methods has been some of the first approaches; however, the requirement of large amounts of labelled data has been a big disadvantage. On the other hand, there are some unsupervised approaches that are employing graphs to represent the text of a document from which the most important nodes are selected as key-phrases. This selection is based on the concept of centrality, i.e. indicators of important nodes within a network.

Centrality is not a new term; it has been used extensively in the study and analysis of social networks and many centrality measures have been developed since then. However, the application of these algorithms specifically to solve the task of key-phrase extraction started just a few years ago; therefore, KE is still an open and unexplored problem in several ways. This task is still far from being solved, note that state-of-the art performance on KE is still much lower than that on many NLP tasks (Liu et al., 2011).

The present work deals particularly with **key-phrase extraction for research publications**. In the case of scientific texts, key-phrases are useful to manage databases, to provide a quick filter for researchers to determine whether a scientific paper is relevant to their interest, to standardise the assignment of key-phrases instead of delegating this task to the authors of a paper, or to automatically assigning reviewers to paper submissions, just to name some relevant applications.

If we take a look at some of the author assigned key-phrases at the top of a paper, we notice that these terms are often **quite specific** and might be shared by a hundred or even a couple dozen of the many thousands of papers that have been published. Besides, as the research literature grows and becomes more specialised, key-phrases will often change over time.

Some previous work in key-phrase extraction attempts to assign key-phrases to research publications using graph-based methods and centrality measures; nevertheless, our understanding is that there has not been an attempt to really deal with the uniqueness of the key-phrases that **characterise** highly specialised documents, such as scientific publications. In this work, we pretend to conduct an evaluation on well-known unsupervised methods for KE and to develop new approaches to improve key-phrase extraction, one of them, called *contrastive betweenness centrality*, is a new centrality measure that attempts to be more specialised for graph-based key-phrase assignment in research publications.

1.1 Objective

Against this background, we ask the following question: Given a real dataset of scientific papers, how does centrality measures behave in the task of key-phrase extraction? Furthermore, is there other method that perform better specifically for scientific papers?

The general purpose of this work is to create a system that automatically identifies the key-phrases that characterise the topics discussed in research papers in the field of Computer Science. A good set of key-phrases stands out the important concepts of original research work, therefore, characteristic key-phrases of a document should not only be a description but a distinctive feature to contrast the document against others.

Differently to previous work, our goal is not only to extract key-phrases, but to **predict what keywords the authors will use to describe a research paper** in order to distinguish its content from a collection of papers in the same field.

Intuitively, what we want is **a short phrase which is highly characteristic of the current paper but not characteristic of many other papers**. Note that inverse document frequency (IDF) of the TF-IDF statistic score does this, and for this reason it is a strong baseline difficult to outperform; but it is still unexplored how to do this in recent graph-based methods for key-phrase extraction. What makes this task challenging is how precisely to formalise this intuition in unsupervised graph-based models.

1.2 Contribution

Our contributions are:

- Construction and experimentation with directed and undirected co-occurrence graphs of noun phrases.
- Ranking the importance noun phrases in a document based on two statistic scores and six centrality measures.
- Performance evaluation of different centrality measures used for key-phrase extraction on a real dataset of Computer Science papers. Comparison of the performance of these measures with TF-IDF, a state-of-the-art baseline.
- Identification of centrality measures that perform well for key-phrase extraction in: full text of papers, abstracts of papers, ranking only noun phrases with very low frequency (key-phrases that occur once or twice in a paper).
- Design and implementation of *contrastive betweenness*, a graph-based method using centrality concepts for key-phrase extraction that pretends to characterise research papers.
- Design and construction of a *document-term graph*, a huge network that comprises the whole collection of papers.
- Implementation of a supervised weighted sum of features to combine the properties of TF-IDF baseline and centrality measures in order to re-score the importance of key-phrases.

- Comparison of the correlation between the different unsupervised key-phrase extraction measures.

1.3 Work structure

The rest of this document is organized as follows:

In Chapter 2, we present a literature review of the state-of-the-art supervised and unsupervised approaches for key-phrase extraction. In the second part of this chapter, we describe the concepts behind the unsupervised graph-based methods in order to support our baseline systems and to justify the design of our approaches.

In Chapter 3, we present the details of our methodology to extract highly characteristic keywords from research papers. We describe the preprocessing of documents to get a list of noun phrases, i.e. candidate key-phrases, the construction of the graph to represent a document and the details of the implementation of the baseline systems. Most important, we explain the conceptual design and the implementation of the three approaches contemplated in this work, which are: *betweenness*, *contrastive betweenness* and *weighted sum of features*.

In Chapter 4, we give a brief review on the popular metrics for evaluating key-phrase extraction systems. Later, we show the results of each experiment performed to evaluate the three approaches and to compare its performance with the centrality measures and the statistic baselines. Finally, more than comparing performance, we get a deeper understanding of the similarities and differences between the different unsupervised KE methods by using Spearman's rank correlation coefficient to test for monotonic relationships between methods.

In the last chapter, we summarise the development of this project and we make conclusions by highlighting the advantages and disadvantages of each system presented along this work. We include some observations related to the improvement areas for key-phrase extraction in the future work.

Chapter 2

Background

The first part of this chapter consists of a literature review on the different methods that have been developed to address the key-phrase extraction problem. It goes from simple methods that involve numerical statistics, to more sophisticated methods that use Machine Learning algorithms, including supervised and unsupervised learning.

In the second part of the chapter, we focus our attention on describing graph theory behind the unsupervised state-of-the-art methods for key-phrase extraction. We emphasise the considerations taken by different authors to choose the nodes, edges and weights to represent a piece of text as a graph; finally, we define the main centrality indicators, which are used to identify the most important key-phrases within a document represented as a graph.

2.1 Previous work on key-phrase extraction

In this literature review, we present a comprehensive overview of the state-of-the-art methods for key-phrase extraction; however, we emphasise on **unsupervised graph-based methods**, which is appropriate for the scope of this work. According to Chen and Lin (2010), we can divide the existing methods into two broad categories:

1. Statistics Approaches and
2. Machine Learning Approaches.

The **statistics approaches** are simple methods, where the statistics of the phrases is used to select the key-phrases in a document. This methods include: N-gram statistics, term frequency or term co-occurrences, among others, being the frequency based TF-IDF algorithm the most popular and frequently used. Generally speaking, these methods have two advantages: (1) they do not require training data, for this reason (2) they are domain-independent. On the other hand, their obvious disadvantage is that in some documents, the important key-phrases may appear just once in the complete text and these methods may inadvertently filter out these phrases (Chen and Lin, 2010).

The **machine learning approaches** are more complex methods that use learning algorithms to determine the key-phrases of a document. **Supervised methods** require a training set of examples for which correct key-phrases have been supplied. These methods learn from the examples to prefer key-phrases over non-key-phrases. These methods include Naive Bayes and SVM, for instance. Although supervised methods can achieve good results with a well-trained model, they have the disadvantage that only are able to extract key-phrases from an specific domain where they were trained. Therefore, the system needs to be re-trained every time that we want to change from one dataset to another. For these reasons, unsupervised methods got the attention of this work. **Unsupervised methods** are domain-independent, therefore, they do not require to re-train the system. Additionally, as they are not based in the statistics of the phrases, they can extract key-phrases that appear just once in a document. On the other hand, it is important to point out that these graph-based state-of-the-art techniques encounter scalability and sparsity problems (Beliga et al., 2014) which were experimented during the development of this work and are detailed in Chapter 3.

Having stated the advantages and disadvantages of the approaches for key-phrase extraction, below we proceed to describe the main research work on each of them.

2.1.1 Statistics methods

These methods are frequency-based, this means, they use the frequency of occurrence of the key-phrases in a document to decide their importance, usually according to the famous TF-IDF score (Manning et al., 2008), which is widely used in information retrieval and text mining (Romero et al., 2012). The first approximation for term weighting in information retrieval was proposed by Salton and Buckley (1988) to use TF-IDF formula to identify relevant documents to a query. Later, this idea was extended by other authors, such as HaCohen-Kerner (2003) who presented one of the first systems to extract key-terms from abstracts and titles of academic papers.

More recent, word association-based approaches assume that semantically similar words tend to occur in similar contexts. Matsuo and Ishizuka (2002) designed an algorithm based on the co-occurrence of the candidate phrases and the frequent phrases of a document, for instance, the number of times that they co-occur in the same sentence. Wartena et al. (2010) makes the assumption that a word occurring in a number of documents on the same topic is more representative of these documents than a word occurring in the same number of documents but scattered over different topics.

2.1.2 Machine learning approaches

A key-phrase extraction system with this approach consists of two stages: (1) preprocess the text to extract a list of candidate key-phrases; and (2) decide which of the candidates are real key-phrases and which are non-key-phrases using supervised (section 2.1.2.1) or unsupervised (section 2.1.2.2) learning.

2.1.2.1 Supervised learning

The first methods developed with supervised learning algorithms, approach the key-phrase extraction as a binary classification problem (Hasan and Ng, 2014). The annotated key-phrases and other non-key-phrases of a document, are used respectively as positive and negative examples to train the classifier, whose final goal is to determine whether a candidate phrase is a key-phrase in an unseen document. Previously, several authors have implemented this approach with different training algorithms, including Naive Bayes (Frank et al., 1999), decision trees (Turney, 2000), bagging (Hulth, 2003), boosting (Hulth et al., 2001), maximum entropy (Yih et al., 2006; Kim and Kan, 2009), multilayer perceptron (Lopez and Romary, 2010), and support vector machines (Jiang et al., 2009; Lopez and Romary, 2010).

The weakness of all these supervised approaches is that a binary classifier classifies each candidate phrase independently, this means, if a candidate phrase c_1 is more representative of a document than another candidate phrase c_2, the binary classifier does not prefer c_1 over c_2. Recall that the goal of key-phrase extraction is to **identify the most representative phrases of a document** (Hasan and Ng, 2014). Motivated by this weakness, Jiang et al. (2009) proposed a method that is able to rank two candidate key-phrases, called Ranking SVM, which takes a pair of phrases as training examples, one key-phrase and one non-key-phrase, and constructs an SVM model based on the training data. Then, with this Ranking SVM model, it is able to sort the candidate key-phrases of an unseen document. This approach outperformed the binary classification approaches mentioned previously.

Krapivin et al. (2010) introduces the use of linguistic knowledge and heuristics for feature selection to improve some of the supervised approaches, such as support vector machines and random forests, for the extraction of key-phrases from scientific publications. The evaluation shows that this approach also outperforms the simple binary classification.

2.1.2.2 Unsupervised learning

Unsupervised methods approach the key-phrase extraction task as a ranking problem. In this area, there is a particular interest for the graph-based methods, which consist in **building a network of candidate phrases and ranking the importance of the nodes using some properties of the graph**, also called centrality measures or centrality indicators.

Since centrality measures captured the attention over other unsupervised methods, researches have used many of them to solve the key-term extraction task; variants of PageRank (Mihalcea and Tarau, 2004; Wan and Xiao, 2008), HITS (Litvak and Last, 2008), and degree, betweenness and closeness (Xie, 2005) have been reported in research literature. Xie (2005) was one of the first authors who built noun phrase graphs to predict phrases that appear in the abstracts of scientific publications.

Boudin (2013) presents a pioneering study in the comparison of various centrality measures for graph-based key-phrase extraction. This research proposes undirected weighted co-occurrence networks that were constructed from preprocessed documents to extract phrases that co-occur within a window. Some of the measures mentioned before are compared one to each other such as degree, closeness, betweenness and PageRank. Closeness shows a better performance in short documents, such as abstracts of the research publications. Also, this work shows that simpler measures, for instance degree, achieves comparable performance to computationally more expensive measures as betweenness.

Lahiri et al. (2014) uses networks of noun phrases to extract keywords from 4 different datasets. Differently to Boudin (2013), in this research paper they use directed and undirected weighted networks. A total of eleven centrality indicators are compared in their results (degree, strength, neighbourhood size, coreness, clustering coefficient, structural diversity index, page rank, HITS hub and authority score, betweenness, closeness and eigenvector centrality). The results obtained suggests that in 2 of the 4 datasets, centrality measures outperform the TF-IDF statistics score. Similarly to Boudin (2013), Lahiri et al. (2014) states that some simpler measures outperform computationally more expensive centrality measures.

Zhou et al. (2013) incorporates linguistic knowledge to build the networks more accurately. It includes a reasonable selection of nodes (selection of words from the text from a linguistic perspective). It associates words semantically in a sentence within a maximum textual distance of 2 in order to build lexical edges. It assigns weights to the edges using Jaccard coefficient, instead of the simplest but less accurate approach of taking the co-occurrence frequency as the edge weight. This approach is compared with three systems for key-term extraction: binary network, simple weighted network and TF-IDF. The results for documents in Chinese suggest that this approach achieves better results in accuracy and recall scores.

Finally, Abilhoa and de Castro (2014) proposes a key-term extraction system for noisy data such as tweets. It represents tweets as graphs and applies the centrality measures to find key-terms in these microblogs. The results show that these unsupervised learning algorithms called graph-based methods, are also able to extract relevant keywords from noisy text data, especially from short texts like tweets.

Much of the fundamentals of the present work have been taken from the research work presented in Boudin (2013) and Lahiri et al. (2014).

2.2 Baseline methods for this work

2.2.1 TF and TF-IDF

Hasan and Ng (2010) demonstrated that TF-IDF algorithm is usually capable to outperform many of the graph-based unsupervised approaches described previously with different datasets. Therefore, in this work we have decided to compare the main centrality measures against two statistics approaches: the weak *term-frequency* (TF) and the much stronger and popular TF-IDF, following the same baselines used in Lahiri et al. (2014).

Whilst TF measures the importance of a term by just counting **the times it appears in a document**, the purpose of TF-IDF is to score **how relevant is a term for a document in a complete collection of documents**, in our case, research publications. For our TF-IDF baseline, we use the same formulation of Nguyen and Kan (2007), where the logarithm is applied to dampen the IDF value:

$$w_{TF \times IDF} = \frac{tf_{w,D}}{|D|} \cdot log \frac{|C|}{df_w} \tag{2.1}$$

Where $tf_{w,D}$ is the TF of a candidate keyphrase, $|D|$ is the length of the document, $|C|$ is the length of the collection of documents, and df_w is the number of documents in the course that contain w.

It is intuitive that with TF, **frequent phrases have a high score**. Considering the IDF part and the entire corpus, **phrases that occur in a few documents are assigned a high score**.

2.2.2 Graph-based methods

Documents can be modelled as graphs where the terms of the document are represented as nodes in a network and the relation between terms is indicated by edges. To assign an importance score to each node, we can perform different types of computations that measure the topological properties of a graph and indicate the relevant nodes according to their criteria.

2.2.2.1 Graph-based text representation

The graph-based representation is one of the most efficient formats to represent the structure of a document and to keep the relationship between phrases or words as accurately as possible (Beliga et al., 2014). For this reason, graph representations are widely discussed in state-of-the-art methods for key-phrase extraction as we reviewed in section 2.1.2.2.

In a sum, there are several different approaches to build a graph from text. The nodes of the graph are the terms of the document, which can be words or phrases; the edges are the relation between those terms and can be weighted or unweighed, directed or undirected. Some of the relevant considerations to model a document as a graph are:

- **Nodes:** Generally, in phrase graphs, nodes are added to the graph restricted with syntactic filters, which select only lexical units called noun phrases (Boudin, 2013). Noun phrases have been chosen by several authors in the field recognising the fact that key-phrases assigned by human experts in practice are usually noun phrases. A noun phrase (NP) is a phrase which has a noun (or indefinite pronoun) as its head word, or which performs the same grammatical function as such a phrase. For example: *This sentence contains two noun phrases.*

- **Edges:** We have established before that edges indicate relation between two phrases, however, these connections can be based on different considerations. According to Sonawane and Kulkarni (2014), edge relation can be based on:

    1. Words occurring together in a sentence, paragraph, section or document (Xie, 2005).

    2. Common words in a sentence, paragraph, section or document.

    3. Co-occurrence in a fixed window of *N* words (Xie, 2005; Boudin, 2013; Lahiri et al., 2014; Abilhoa and de Castro, 2014).

    4. Semantic relation of two words: synonyms, antonyms, words with same spelling but different meaning, etc. (Zhou et al., 2013).

- **Weights:** Edges in a graph can be considered weighted or unweighed. Weights are recommended when it is desired to measure **the strength of the relationship** between two nodes. Usually, it is more common to find weights assigned in co-occurrence edges than in semantic edges. In co-occurrence edges the weight is measured as the co-occurrence frequency of the terms they connect (Boudin, 2013). Zhou et al. (2013) uses Jaccard coefficient to consider not only the co-occurrence frequency, but also the frequency of both phrases separately. Mihalcea and Tarau (2004) measures weights as the distance between word occurrences.

Unfortunately, there is not a standard graph model for representing text in the most effective way, hence, depending on the application, different graph-based representations can be experimented and used (Sonawane and Kulkarni, 2014).

2.2.2.2 Centrality measures

The centrality measures are discriminative properties of the **importance of a node** in a graph (Beliga et al., 2014). These measures are widely used in graph-based methods for keyword extraction, and they are detailed in Boudin (2013); Beliga et al. (2014); Lahiri et al. (2014). In this section, we describe the basic measures that we use to compare the methods developed in Chapter 3. For this work, N is the number of nodes in a graph and K is the number of edges. In weighted graphs, every link connecting two nodes i and j has an associated weight which is a positive integer w_{ij}.

Table 2.1 lists all the centrality measures used in our study and where to find them in previous research work. Definitions of the centrality measures follow:

- **Degree centrality:** It is the number of edges incident to a node i, defined as k_i. It can be normalised by dividing the value by the maximum possible degree $N-1$:

$$dc_i = \frac{k_i}{N-1} \tag{2.2}$$

- **Strength centrality:** It is the sum of all the weights of the links j incident to a node i.

$$sc_i = \sum_j w_{ij} \tag{2.3}$$

- **Closeness centrality:** It is the inverse of farness, which is the sum of the shortest distance between a node i and all the other nodes j. We define σ_{ij} as the shortest path, this is the path between two nodes in a graph such that the sum of the weights of the edges on the path is minimised. In the case of unweighed graphs, the shortest path is the minimum length with all edge weights equal, e.g. all set to 1. Then, we have the normalised formula of closeness defined as:

$$cc_i = \frac{N-1}{\sum\limits_{i \neq j} \sigma_{ij}} \tag{2.4}$$

Centrality measure	**Previous research work applied to keyphrase extraction**
Degree	(Xie, 2005; Boudin, 2013; Lahiri et al., 2014; Beliga et al., 2014)
Strength	(Lahiri et al., 2014; Beliga et al., 2014)
Closeness	(Xie, 2005; Boudin, 2013; Lahiri et al., 2014; Beliga et al., 2014)
PageRank	(Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Lahiri et al., 2014; Beliga et al., 2014)

Table 2.1: Graph centrality measures used in this work.

- **PageRank:** PageRank is a method for capturing the intuition that important nodes are those that link to other important nodes. In the simplest version of Page et al. (1999), PageRank of i is defined as the sum of the normalised rank of all the nodes $j \in B_i$ with a link pointing to i. The normalised rank is the PageRank of j divided by the number of edges from j that is:

$$R_i = c \sum_{j \in B_i} \frac{R_j}{N_j} \tag{2.5}$$

Where c is a factor used for normalisation so that the PageRank values sum to a constant. Equation 2.5 is recursive. It can start with any set of ranks and should keep iterating the computation until it converges, this is, ranks find a fixed value.

Chapter 3

Method

In this chapter, we use the foundations of Chapter 2 to detail the methodology of three different approaches to extract key-phrases. Firstly, we detail the pre-processing stage and the construction of the graphs. Subsequently, the implementation of the baseline methods is reported. Finally, we detail the conceptual design and the implementation of the three approaches contemplated in this work. The first simple approach is the implementation of the betweenness centrality indicator in a document graph; the second approach is a novel modification of the betweenness centrality formula, called *contrastive betweenness*, that intends to include the information of all the papers of the dataset in a single graph, which has been called *document-term graph*; in the third approach, we experimented with a supervised weighted sum to combine as features the TF-IDF score and the betweenness centrality measure.

3.1 Preprocessing

Our methodology is based on that proposed by Boudin (2013) and has four steps: (1) pre-process the documents to get the candidate key-phrases of each paper; (2) build a graph using the candidate key-phrases and its relations; (3) compute a centrality indicator to score the importance of the nodes of the graph; and (4) rank the key-phrases according to this score and choose from the top.

The preprocessing stage was developed using NLTK library in Python. The aim of this stage is to get candidate key-phrases, which are the noun phrase chunks from the text of each document. Based on the research of Krapivin et al. (2010); Boudin (2013);

Lahiri et al. (2014); Beliga et al. (2014), we developed a set of preprocessing steps that are described below:

1. We use a **boundary detector** to divide the input text into sentences.
2. For each sentence, we apply **tokenisation** to break it into words and symbols.
3. **POS tagging**.
4. **Parsing**.
5. **Chunking** to identify the constituents noun groups. We apply a heuristic based on the linguistic analysis made in Krapivin et al. (2010) specifically for the expert assigned key-phrases of the dataset we are using. This analysis concludes that the overwhelming majority of the assigned key-phrases are noun phrases, therefore, it recommends to filter the chunks by chunk type leaving only noun phrase (NP) chunks.
6. We **lowercase** the list of noun phrases obtained in the last step.
7. **Stemming** and **lemmatisation** of the words of each noun phrase to avoid same words written in different forms.
8. Finally, we apply other heuristic taken from Lahiri et al. (2014), which **filters out** single-world noun phrases that are **stopwords** and **noun phrases with more than five words** because usually authors does not use very large key-phrases to describe their papers. These filters limit the dimensions of the graph that we will construct in the next step.

The results of our preprocessing stage can be visualised in Table 3.1 that shows an example sentence and the list of candidate key-phrases after preprocessing this sentence. The result is the extraction of noun phrases that are linguistically meaningful to serve as candidate key-phrases. For the entire dataset, the preprocessing stage takes around eight hours to obtain the candidate key-phrases of the 2304 research publications.

After getting the candidate key-phrases of all the documents of the dataset, we compute the percentage of expert assigned key-phrases that we caught in the candidates. We find that, after preprocessing, 71.82% of the expert assigned key-phrases are in the candidate key-phrases. The analysis of the expert assigned key-phrases in our dataset will be explained when we describe the corpus in section 4.1.1; but the results of this analysis indicate that the maximum recall we can get after preprocessing the docu-

Sentence	A general model of software development environments that consists of structures, mechanisms, and policies is presented. The advantage of this model is that it distinguishes intuitively those aspects of an environment that are useful in comparing and contrasting software development environments.
Candidate key-phrases	gener model, softwar develop environ, structur, mechan, polici, advantag, model, aspect, environ, compar, softwar develop environ

Table 3.1: Candidate key-phrases extracted by the preprocessing stage.

ments is around 75% because the remaining 25% of the expert assigned key-phrases does not appear textually in the documents. Hence, we are only missing 3.18% of the key-phrases after preprocessing the documents.

3.2 Graph construction

After preprocessing the dataset, we get a set of unique noun phrases for each document which are the candidate key-phrases of it. Therefore, we reproduce the state-of-the art methods mentioned in section 2.1.2.2 by building two co-occurrence networks (a directed and an undirected) for each document. A co-occurrence network is a graph where **the nodes are unique candidate key-phrases and the edges are links between two nodes that occur within a window of each other** (Ferret, 2002). Then, our graphs are built with the next characteristics:

- **Nodes:** Each node represent a candidate key-phrase from the list of **unique noun phrases** that was created during the preprocessing stage of a document.
- **Edges:** In our experiments, we used both weighted directed and weighted undirected graphs because some centrality measures makes sense with directed graphs and others with undirected. In the case of undirected graphs, undirected edges link two nodes when a pair of candidate key-phrases co-occurs within a window. For the directed graphs, we use the same approach where a direction was set following the natural flow of the text in forward and backward direction (Son-

awane and Kulkarni, 2014). Following Mihalcea and Tarau (2004); Wan and Xiao (2008); Boudin (2013); Sonawane and Kulkarni (2014), we set the **co-occurrence window size to 10 phrases** for all our experiments; furthermore, we considered a simplified version of these graphs where all self-loops were removed in order to save computational cost.

- **Weights:** We chose a simple approach to establish weights. The weight of an edge is proportional to the overall **co-occurrence frequencies** of the corresponding pair of candidate key-phrases within a document.

Graph construction was implemented with a script in Python and consists in reading the list of noun phrases of each document and pairing all the unique noun phrases that exists within a window of ten phrases. This window goes over the entire list of noun phrases. At the end, the number of repeated pairs are counted to assign the corresponding weight to the edge that links each pair. The unique list of edges and their weights are written in a **.txt* document. Later, we used Networkx library in Python to create a directed or an undirected graph from this list of edges.

To illustrate the result of the graph construction that was described above, we took the example sentence of Table 3.1 in last section. In Figure 3.1, we display a sample directed graph that results from this sentence. In Figure 3.2, we display a sample undirected graph that results from the same sentence.

3.3 Baselines

The main baseline methods for this work are the two statistical methods we presented in section 2.2.1: a weak baseline, TF, and a much stronger and popular algorithm, TF-IDF. Both baselines were implemented in Python scripts. The TF function simply counts the frequency of the candidate key-phrases in a document and ranks them from higher to lower TF score. The TF-IDF implementation is more complex. It implements the equation 2.1 that we showed in section 2.2.1. For implementing this equation, it is necessary to have the complete dataset, since the IDF part of the equation requires to count the number of documents where each key-phrase appears.

On the other hand, centrality measures (degree, closeness and PageRank) were computed directly with the Networkx library. Degree and closeness were calculated from the undirected graph, whilst PageRank was computed from the directed version of the

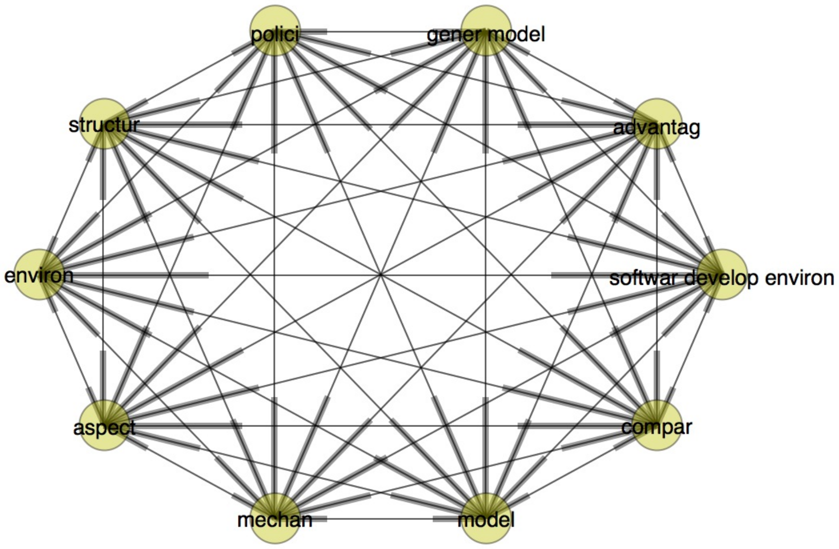


Figure 3.1: Sample directed graph drawn from sample sentence in Table 3.1.

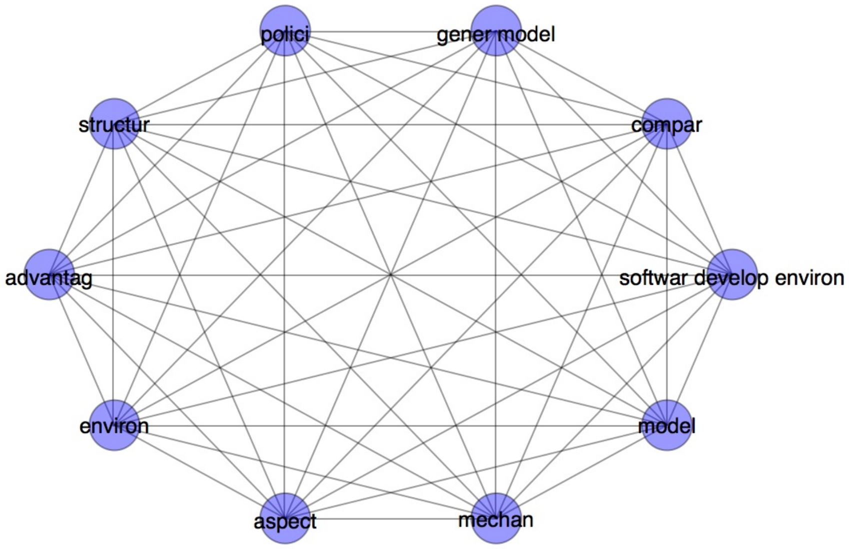


Figure 3.2: Sample undirected graph drawn from sample sentence in Table 3.1.

Expert assigned key-phrases	TF	TF-IDF	Degree	Strength	Closeness	PageRank
program environ softwar develop environ languag center environ method-bas environ sociolog metaphor structure-ori environ toolkit environ	polici	polici	environ	polici	environ	polici
	environ	famili model	structur	environ	structur	environ
	structur	sde	polici	structur	polici	structur
	mechan	individu model	mechan	mechan	mechan	mechan
	tool	infus	exampl	tool	tool	tool
	exampl	environ	tool	exampl	exampl	exampl
	model	softwar develop environ	model	model	model	model
	system	citi model	system	system	softwar develop environ	system
	develop	mechan	develop	develop	system	develop
	problem	istar	softwar develop environ	problem	develop	problem

Table 3.2: Top ten key-phrases extracted from document #5 of our dataset with different methods.

graph due to the PageRank algorithm itself was designed for directed edges. Networkx has not implemented and algorithm to compute the centrality measure of strength, therefore, it was implemented in an additional script in Python and was calculated from the undirected graph.

In Table 3.2, we show the top ten key-phrases obtained for document #5 of our dataset, which were ranked according to TF, TF-IDF, degree, strength, closeness and PageRank.

3.4 Betweenness centrality measure

The first approach proposed in this work is to compute betweenness centrality measure to score the importance of the candidate key-phrases in the graphs that we constructed in section 3.2. Betweenness centrality of the node i is defined as the number of times it acts as a bridge along the shortest path between two other nodes of the graph (Boudin, 2013). Let σ_{jk} be the number of shortest paths between nodes j and k and let $\sigma_{jk}(i)$ be the number of those paths that pass through i, and there are N nodes in the graph, then the normalised betweenness centrality of i is defined as:

$$bc_i = \frac{\sum\limits_{i \neq j \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}}}{(N-1)(N-2)} \tag{3.1}$$

Following the above definition, betweenness is a measure of **how often a node is located on the shortest path between other nodes in the graph**. We have already defined previously in this work a shortest path as the path between two nodes in a graph such that the sum of the weights of the edges on the path is minimised. In the case of unweighed graphs, the shortest path is the minimum length with all edge weights equal, e.g. all set to 1. Assuming that the transference of the communication in a document is through the shortest paths, thus betweenness measures **the degree to which a node acts as a point of control in the communication** (Leydesdorff, 2007). This is, if a node with high betweenness score is deleted from the graph, it would fall apart into otherwise coherent clusters.

For this reason, we are interested in betweenness to find the key-phrases that characterise research publications. In previous work such as Boudin (2013); Abilhoa and de Castro (2014), closeness has been found to be an appropriate measure to find the key-phrases in short texts. However, closeness favours nodes belonging to dense groups in the graph, but in long texts, where there is a large yet very sparse network, we expect that important nodes does not belong to one of the dense groups of the graph, but relates them. This is, **the nodes with high betweenness score are key concepts that relates the different sections or topic segments of a research publication**.

For this work, betweenness centrality measure was computed in Python with the function developed in Networkx library and was calculated from the undirected version of the graph that was built in section 3.2. Networkx uses the fast algorithm by Brandes (2001) for the computation of betweenness measure. This algorithm requires $O(N+K)$ space and has complexity $O(NK)$, N being the number of nodes and K the number of edges in the network.

In Table 3.3, we show the top ten key-phrases obtained for document #5 of our dataset, which were ranked according to betweenness centrality score.

3.5 Contrastive betweenness centrality measure

The main approach of this work is this novel method that we developed based on the concept of betweenness centrality that we have explained previously. Betweenness measures the control that a node exerts in the graph. When a node that exerts control in the communication is removed from the graph, it could generate two clusters (subnetworks) such that points in one network are not reachable from the other (Stephenson and Zelen, 1989).

Expert assigned key-phrases	Betweenness
program environ softwar develop environ languag center environ method-bas environ sociolog metaphor structure-ori environ toolkit environ	environ structur tool polici mechan exampl model softwar develop environ program system

Table 3.3: Top ten key-phrases extracted from document #5 of our dataset with betweenness centrality.

With this in mind, our approach to extract key-phrases that truly characterise a research paper is as follows:

- If a node is a **good bridge within the document**, then, it is a key-phrase; if this node is also a **good bridge within the collection of documents**, then, this key-phrase is "interdisciplinary" (shared between many documents) and it **does not characterise** the specific topics of a research paper.
- Conversely, if a node is a **good bridge within the document**, then, it is a key-phrase; but if this node is **not a good bridge within the collection of documents**, then, this key-phrase is exclusive of a document and it **may characterise** the specific topic of this research paper.

Supported by this reasoning, we developed a new method to predict the key-phrases that a human expert would use to characterise the content of a research paper. As we will be developing a graph-based method, we follow the same steps explained at the beginning of this chapter, but with the next variations: (1) a construction of a **complete graph including all the documents in our collection**; (2) extend the formula of betweenness centrality to determine the importance of each node by **contrasting its betweenness score within the document against its betweenness score within the corpus**; (3) for each paper, rank its key-phrases according to the new contrastive betweenness score. The details of the steps of graph construction and the new contrastive betweenness centrality are explained below.

3.5.1 Document-term graph construction

For this approach, we build a graph with nodes represented as a pair (d, v), where d is the document and v is a candidate key-phrase that occurs in d. The edges of this graph link key-phrases within a window size of ten. The novelty is the addition of **edges across documents** when they share common candidate key-phrases. This graph has been called *document-term graph* for this work. The steps to build this network are described as follows:

1. Extract the candidate key-phrases of each document and draw edges that represent the co-occurrence relations within a window of ten phrases. Build **a graph per document** exactly as section 3.2.
2. Look for the candidate key-phrases that **occur in more than one document**.
3. Draw **edges across documents** to link the nodes with same v (candidate key-phrase), but different d (document). Assign a weight to these edges according to the probability of passing from one document to another given that they share a candidate key-phrase. This is,

   We define the number of documents in the dataset that contain the candidate key-phrase v_1 as:

   $$N(v_1) = df_{v_1} \tag{3.2}$$

   Therefore, the probability of moving to a document d_2, given that we are in the key-phrase v_1 of the document d_1 is:

   $$P(move\ to\ d_2|(d_1, v_1)) = \frac{N(v_1) - 1}{N(v_1)}; d_1 \neq d_2 \tag{3.3}$$

   Which is precisely the **weight of the edge**.

In Figure 3.3 we exemplify the structure of a simplified document-term graph for a small collection of three documents.

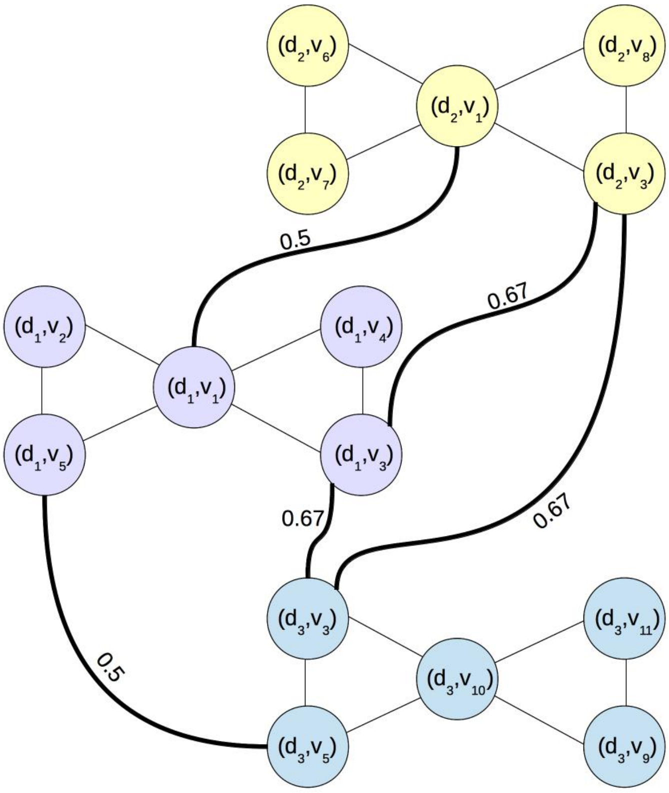


Figure 3.3: Sample graph drawn to illustrate the construction of the document-term graph in this work.

3.5.2 Contrastive betweenness score

To capture our intuition about the possibility to use betweenness centrality to find key-phrases that characterise a research paper, explained at the beginning of section 3.5, we formulate an extension to the original equation of betweenness. In equation 3.1, we use betweenness centrality to measure h**ow easy is to get to the rest of the paper from a key-phrase**. Currently, our new theory is that we can add a consideration about **how difficult is to escape the document from a key-phrase** by subtracting the betweenness centrality obtained with the document-term graph, as follows,

Let (d, v) be a node pair. Given (d, v), consider all pairs $(d_1, v1)$, $(d_2, v2)$. Compute *contrastive betweenness* of the node (d, v) as below:

$$bc_{(d,v)} = \frac{\sum\limits_{v_1 \neq v_2 \neq v} \frac{\sigma_{(d,v_1)(d,v_2)}((d,v))}{\sigma_{(d,v_1)(d,v_2)}}}{(N_d - 1)(N_d - 2)} - \frac{\sum\limits_{\substack{(d_1,v_1) \neq (d_2,v_2) \neq (d,v) \\ d_1 \neq d_2}} \frac{\sigma_{(d_1,v_1)(d_2,v_2)}((d,v))}{\sigma_{(d_1,v_1)(d_2,v_2)}}}{(N-1)(N-2)} \tag{3.4}$$

Where N_d is the number of nodes in the document graph and N is the number of nodes in the document-term graph.

3.5.3 Implementation

This approach was implemented with a script in Python and consists in reading the list of candidate key-phrases of each document and pairing all the candidate key-phrases that occur in two different documents. At the end, the equation 3.3 is calculated to assign the corresponding weights to the edges across documents . This list of edges (pairs of: document, candidate key-phrase) and their weights are written in a **.txt* document. Later, we used Networkx library in Python to create an undirected graph from this list of edges.

For this work, betweenness centrality measure was computed in Python with Networkx library. After computing the betweenness of the document-term graph, we opened the results of the betweenness measure calculated previously for each document graph, and we subtracted the betweenness measure calculated from the huge document-term graph. In other words, we implemented the formula in equation 3.4 to get the **contrastive betweenness score** of the nodes in each document.

In Table 3.4, we show the top ten key-phrases obtained for document #5 of our dataset, which were ranked according to the contrastive betweenness formulation.

However, it is important to mention that during the implementation of this approach, we encountered problems of **scalability**. The dimensions of the document-term graph grew considerably with the edges across documents to $345,214,907$ edges. Thus, with standard computational resources, **it was not possible to compute the betweenness for the huge document-term graph**. Hence, we had to implement different heuristics to reduce the dimensions of this document-term graph:

1. We **reduced the dataset** taking only 10% and 5% documents from the original collection.

2. We extracted the **key-phrases only from the abstracts** of the papers in order to reduce the nodes of the graph.

3. We cut around 50% of the dimensions of the graph using **only the candidate key-phrases with a TF greater than one**, as a consequence of Zipf's law (Li,

Expert assigned key-phrases	Contrastive betweenness
program environ softwar develop environ languag center environ method-bas environ sociolog metaphor structure-ori environ toolkit environ	environ structur tool polici mechan exampl model softwar develop environ program system

Table 3.4: Top ten key-phrases extracted from document #5 of our dataset with contrastive betweenness centrality.

1992).

4. We used **only the candidate key-phrases with a low TF** because according to our study of the expert assigned key-phrases in section 4.1.1, the majority of the key-phrases that we want to extract **occur just one or twice** in the text.

Some of these heuristics worked to compute this approach and the results will be shown in Chapter 4.

3.6 Weighted sum

The last approach is not an unsupervised graph-based approach. It is a **supervised weighted sum to combine features** of different methods for key-phrase extraction. Since TF-IDF is a strong baseline difficult to outperform, but still the results obtained from it are not enough for the task, our purpose is to beat this baseline by **improving this score with graph-based measures**.

To study the suitability of this approach, we propose a simple **weighted sum of TF-IDF score and betweenness centrality score**. What we want is to find a parameter α such that each score contributes in some proportion to **re-score and re-rank** the candidate key-phrases:

$$weighted\ score = \alpha \times w_{TF \times IDF} + (1 - \alpha) \times bc_i \tag{3.5}$$

For this purpose, we scaled first the TF-IDF and the betweenness scores in order to have their values in the same range (between 0 and 1). Then, we split the dataset into two equivalent parts: 50% documents for the training set and 50% documents for the testing set. Validation set is not necessary in this case since there is only one parameter we are adjusting. With the training set, we tried five different values of $\alpha \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$. We determined our best α value and we applied it to compute the weighted score in our testing set.

Our assumption is that the combination of these two features would be a right choice due to TF-IDF is a strong score for the key-phrase extraction task and can be enriched by the mentioned properties of betweenness. To make the features selection, we also studied the correlation between TF-IDF and the centrality measures, in order to choose the least correlated centrality and avoid redundancy. The results of this study and the other two approaches are analysed in Chapter 4.

Chapter 4

Evaluation

In this chapter, we describe the dataset that we employed to evaluate our approaches followed by the conventional metrics for evaluating the performance of key-phrase extraction systems (precision, recall, f-score and precision-recall curves). The second part of the chapter shows the numerical results obtained and displays the resulting P-R curves for each of the three approaches that were described and implemented in Chapter 3. In this part, we also make an analysis on how the results compare to the baseline methods. Finally, we desire to compare the methods not just for their performance; we pretend to get a deeper understanding of the similarities and differences between methods by using Spearman's rank correlation coefficient.

4.1 Dataset description

The dataset was obtained from Krapivin et al. (2008) consisting on 2304 research publications of the Association for Computing Machinery (ACM) in the Computer Science domain in 2003 - 2005. The full texts of the papers were downloaded from CiteSeerX digital library. Subsequently, the text was cleaned by removing formulas, tables, figures and possible Latex mark-up. A pair of files was generated for each paper: $< id >$ *.txt* with full text, and $< id >$ *.key* with key-phrases each on a new line.

The text file has indicators to separate different parts of the paper such as title and abstract, which will be useful to enable the extraction based just on a part of the paper. The key-phrases of the key file were manually assigned by human experts in the domain. Each paper in the dataset contains at least one expert assigned key-phrase that occur in the text. Particularly in this dataset, the average for each paper is about **3 unique expert assigned key-phrases that occur in the full text** of the publication (Krapivin et al., 2010).

After a linguistic analysis of the expert assigned key-phrases, Krapivin et al. (2010) confirms that **the overwhelming majority of key-phrases are noun phrases**. Therefore, we conclude it is a reasonable to represent the documents of this dataset as a graph of noun phrases to apply the methods of key-phrase extraction.

4.1.1 TF analysis of key-phrases

As we mentioned before, in this dataset there is an average of 3 expert assigned key-phrases that occur in the text of a document. However, it is more accurate to compute a number to indicate clearly the percentage of expert assigned key-phrases that appear in the text of the papers in this dataset, and the percentage of expert assigned key-phrases that does not appear and, therefore, we will not be able to find through this methodology. This number will be helpful for two reasons: (1) to have a realistic idea of the maximum percentage of expert assigned key-phrases that we can get after preprocessing the documents (finding the noun phrases), and (2) to get a realistic idea of the maximum value of the recall that we can get after performing any of the proposed key-phrase extraction methods of this work.

A script in Python was developed to open the key file $< id >$ *.key* of each research publication, to extract the list of expert assigned key-phrases and to look for the term frequency (TF) of each of them in the full text of the document $< id >$ *.txt*. The resulting histogram (normalised) is displayed in Figure 4.1.

The results indicate that about **25% of the expert assigned key-phrases does not appear at all in the full text** of the documents, therefore, the maximum recall we can get after preprocessing the documents and performing the key-phrase extraction is around 75%. Additionally, it is interesting to notice that about **10.7% of the expert assigned key-phrases (a majority) appear just once in the documents**. This justify why we are implementing graph-based unsupervised methods instead of statistics methods, which rely highly on the frequency of the key-phrases as we mentioned in Chapter 2.

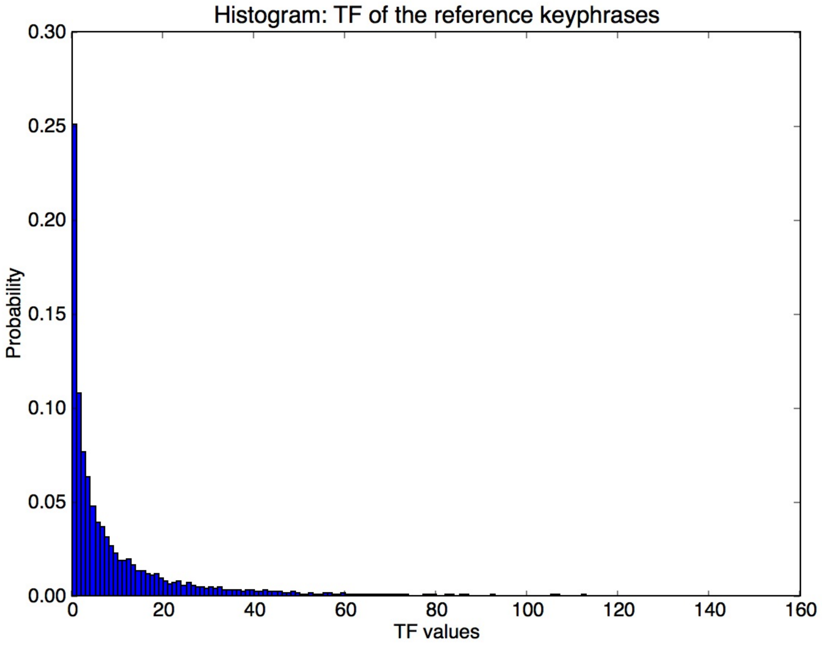


Figure 4.1: Histogram of TF of the expert assigned key-phrases of the dataset in this work.

4.2 Evaluation metrics

The design of a system for evaluating key-phrase extraction is not an easy task. The most widespread evaluation approach consists in finding the **exact match between the gold standard and the extracted key-phrases**, and then, score the output using precision and recall (Hasan and Ng, 2014; Lahiri et al., 2014). According to Lahiri et al. (2014); Haddoud et al. (2015), the state-of-the-art performance shows that key term extraction is a hard problem in general with low f-measures below 20%. Therefore, we have applied lowercase, stemming and lemmatisation stages to our reference expert assigned key-phrases to reduce the number of exact mismatches. Consistent with Hasan and Ng (2010); Boudin (2013); Lahiri et al. (2014), we report the performance of our different key-phrase extraction methods with **precision, recall and f-score at the top 10 key-phrases**, and precision-recall (P-R) curves. Following Boudin (2013), to generate the curves we vary the number of extracted key-phrases from 1 to the total number of candidate key-phrases.

4.2.1 Precision

We calculate precision as the **proportion of extracted key-phrases that match the expert assigned key-phrases** (Haddoud et al., 2015).

Mathematically, we define precision as follows:

$$precision = \frac{|(reference\ keywords)\bigcap(extracted\ keywords)|}{(extracted\ keywords)} \tag{4.1}$$

4.2.2 Recall

The recall is computed as the **proportion of the expert assigned key-phrases that were extracted by our system** (Haddoud et al., 2015).

Mathematically, we can define recall with the next formulation:

$$recall = \frac{|(reference\ keywords)\bigcap(extracted\ keywords)|}{(reference\ keywords)} \tag{4.2}$$

4.2.3 F-score

This is a measure that **combines precision and recall** metrics. Usually, this is calculated as the harmonic mean of both:

$$F - score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.3}$$

This also receives the name of F1 measure because recall and precision are evenly weighted.

4.2.4 P-R Curve

Typically, precision and recall are inversely related, this is, as recall increases, precision falls and inversely. A balance between these two characteristics in any NLP task should be achieved, therefore, P-R curves are useful to make a fair comparison between different systems or methods. There is **not an unique way** to interpret these curves. Depending on the requirement (high precision at the cost of recall, or high recall with lower precision), an appropriate algorithm can be chosen.

The following sections 4.3, 4.4 and 4.5 display the P-R curves to compare the baseline methods with the three approaches for key-phrase extraction proposed in this work. Also, we present in tables the values of precision, recall and f-score at the top 10 key-phrases.

4.3 Evaluation of betweenness centrality measure

To evaluate the results of this approach, we have two variants. Both experiments were made with the complete dataset of 2304 documents. The first variant consisted of extracting key-phrases from the **full text** of the research publications and the second variant consisted of extracting key-phrases **exclusively from the abstract** section.

Table 4.1 presents the performance of each statistics method and centrality measure on our dataset when extracting key-phrases from the full text of scientific publications. The highest score of each evaluation measure has been highlighted. Overall, we observe that the best results are obtained by TF-IDF, our strong statistics baseline; while TF, as expected, yields the worst performance on the dataset. Specifically among centrality measures, betweenness obtains the best results, supporting our assumption in section 3.4 that this measure would work well in long text of research publications.

To get a better understanding of the performance for each centrality measure, we report in Figure 4.2 their P-R curves. Again, we observe that the best centrality measure for our dataset is betweenness, but unfortunately it is not able to outperform TF-IDF for key-phrase extraction. We note that the maximum recall is about 71.82%. Interestingly, TF and strength achieve similar performance, which means that our implementation of strength is proportional to term-frequency. TF can therefore advantageously replace strength for key-phrase extraction.

Table 4.2 presents the performance of each statistics method and centrality measure on our dataset when extracting key-phrases only from the abstracts of the scientific publications. The highest score of each measure has been highlighted. Again, we observe that the best results are obtained by TF-IDF method. Specifically for the centrality measures, degree obtains the best results in first place and closeness in second place.

	P (%)	**R (%)**	**F1 (%)**
TF	4.5770	10.0660	6.2927
TF-IDF	**8.5661**	**18.9984**	**11.8081**
PageRank	4.8453	10.7237	6.6747
Degree	5.4608	12.1633	7.5376
Betweenness	6.5656	14.8780	9.1107
Closeness	5.8199	13.0003	8.0403
Strength	4.5059	9.8854	6.1903

Table 4.1: Performance of statistics methods and centrality measures in terms of precision, recall and f-score at the top 10 key-phrases of the dataset. This key-phrase extraction experiment was performed using the full text of the research publications of the complete dataset.

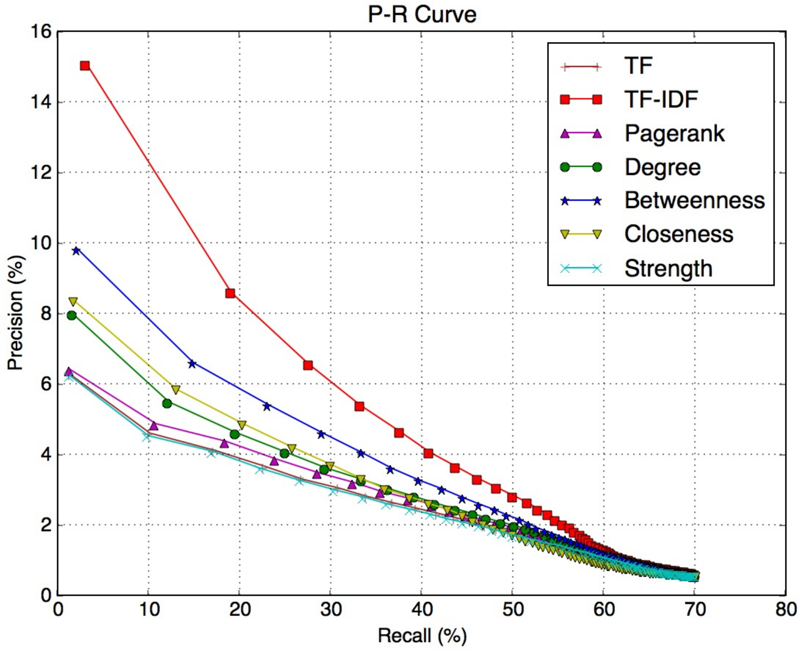


Figure 4.2: Results of the betweenness centrality performance compared to the baseline statistics methods and other centrality measures. This key-phrase extraction experiment was performed using the full text of the research publications of the complete dataset.

These results were as expected; recall we have stated before that Boudin (2013); Abilhoa and de Castro (2014) stated that closeness is a good key-phrase extraction method for short texts (in this case, abstracts). Also, we have presented the results of Boudin (2013); Lahiri et al. (2014), where authors conclude that simpler centrality measures as degree are able to outperform computationally more expensive measures as closeness and betweenness.

	P (%)	**R (%)**	**F1 (%)**
TF	6.5439	13.9227	8.9031
TF-IDF	**7.8708**	**16.3397**	**10.6240**
PageRank	6.2872	13.2166	8.5209
Degree	6.1568	12.8170	8.3180
Betweenness	6.0976	12.7285	8.2453
Closeness	6.0621	12.6247	8.1910
Strength	6.3424	13.3517	8.5998

Table 4.2: Performance of statistics methods and centrality measures in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using exclusively the abstract of the research publications of the complete dataset.

To get a broader viewing and understanding of the performance for each centrality measure, we report in Figure 4.3 their precision-recall curves for this experiment. In this plot we observe that, overall, the best method is TF-IDF followed by the weak TF baseline. On the other hand, there is no centrality measure which overall performs best. We note that the maximum recall is about 27.74%, which is low compared when we used the full text of the papers in the last experiment.

For this approach, we can conclude that, in general, **betweenness is the best centrality measure to extract key-phrases from our dataset when we are employing the full content of our documents**, which is better than extracting key-phrases from the abstracts because our recall increases dramatically. However, **betweenness centrality is not helping to outperform the TF-IDF baseline**, which is a statistics measure that additionally to better performance, implies lower computational cost and time.

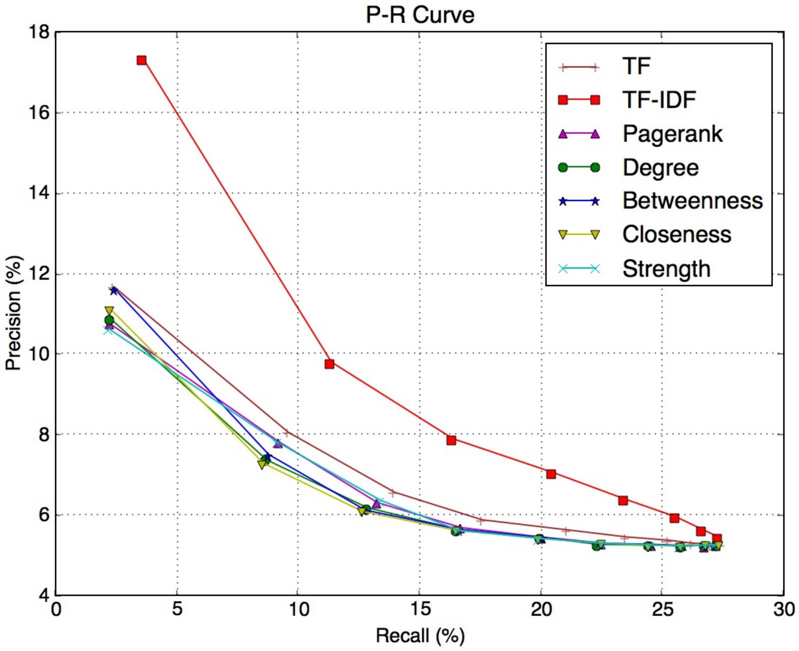


Figure 4.3: Results of the betweenness centrality performance compared to the baseline statistics methods and other centrality measures. This key-phrase extraction experiment was performed using exclusively the abstract of the research publications of the complete dataset.

4.4 Evaluation of contrastive betweenness centrality measure

To evaluate the results of this approach, we have four variants in the experiments. As we have stated in Chapter 3, standard computational resources were not enough to compute properly this approach with the complete dataset due to the dimensions of the resulting document-term graph. Hence, we had to implement different heuristics to reduce these dimensions and the computational cost of calculating the contrastive betweenness centrality measure. The experiments were developed with the next features:

1. We reduced the dataset taking only 5% documents from the original collection.
2. We took 10% documents from the original collection and extracted the key-phrases only from the abstracts.
3. We took 5% documents from the original collection and built the graphs using only the candidate key-phrases with a TF greater than one.

4. We took 5% documents from the original collection and built the graphs using only the candidate key-phrases with a low TF.

Table 4.3 presents the performance of each statistics method and centrality measure (including the contrastive betweenness) on our dataset when extracting key-phrases from 5% of the scientific publications. The highest score of each evaluation measure has been highlighted. We notice that the best results are obtained using the TF-IDF baseline, followed by degree centrality, a very simple measure. On the other hand, there is no complex centrality measure like betweenness or the contrastive betweenness which performs better. However, note that betweenness and its contrastive version could be proportional, since they have the same values of precision, recall and f-score in this threshold (at top 10 key-phrases).

	P (%)	**R (%)**	**F1 (%)**
TF	4.3478	6.2318	5.1220
TF-IDF	**7.5098**	**9.7184**	**8.4725**
PageRank	5.1383	6.5257	5.7495
Degree	5.9288	8.0474	6.8275
Betweenness	5.9288	7.4644	6.6086
Closeness	5.9288	7.6851	6.6937
Strength	3.1620	2.5290	2.8103
Contrastive betweenness	5.9288	7.4644	6.6086

Table 4.3: Performance of statistics methods and centrality measures in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using the full text of 5% research publications of the dataset.

To get a broader viewing and understanding of the performance for each centrality measure, we report in Figure 4.4 their precision-recall curves for this experiment. In this graph it is evident that, in general, the best method is TF-IDF. Moreover, there is no centrality measure which overall performs best. We can appreciate contrastive betweenness (a.k.a. modified betweenness) curve above the rest of the other centrality curves, but its performance is not notably better. Also, it is clearer with the P-R curves that betweenness and the contrastive version are not performing exactly in the same way, as could be supposed for the results of Table 4.3. Note that TF-IDF and contrastive betweenness curves have the same starting value; since we start drawing the curve at top 1 key-phrases, this is an indicator that they usually select the same key-phrase in first position. In this case, the maximum recall is about 70.25%.

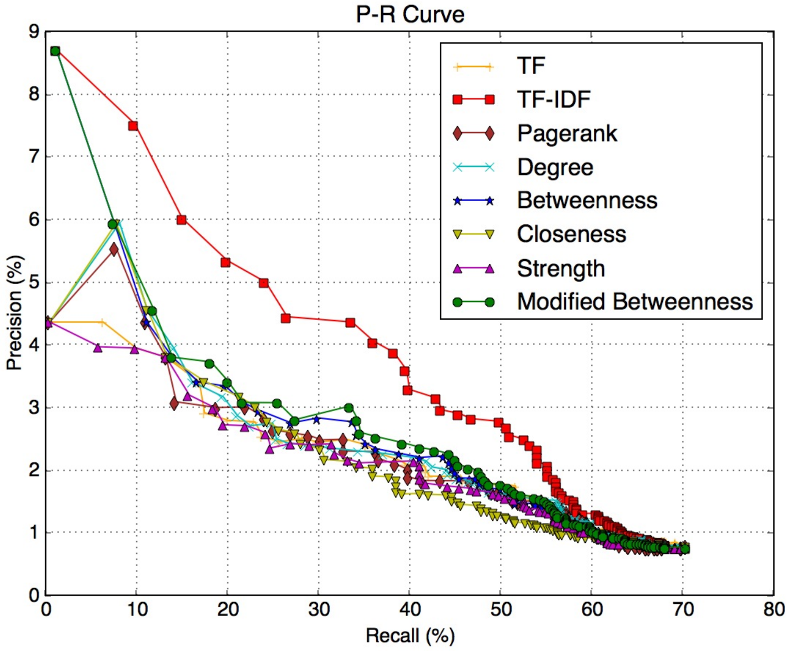


Figure 4.4: Results of the contrastive betweenness (a.k.a. modified betweenness) performance compared to the baseline statistics methods and other centrality measures. This key-phrase extraction experiment was performed using the full text of 5% research publications of the dataset.

Table 4.4 contains the performance of each statistics method and centrality measure (including the contrastive betweenness) on our dataset when extracting key-phrases from the abstracts of 10% of the papers. The highest score of each evaluation measure has been highlighted. As we have noticed in the last experiment, the best results here are also obtained using the TF-IDF baseline, this time followed by strength centrality and contrastive betweenness. However, in this experiment is not clear which centrality measure has a predominant performance over the rest.

For this reason, we illustrate the performance for each centrality measure in Figure 4.5 with their precision-recall curves for this experiment. The results indicate that the best performing method is TF-IDF. Moreover, there is no centrality measure which overall performs best. We can appreciate that the contrastive betweenness (a.k.a. modified betweenness) curve behaves better than other centrality measures at low recall, but as soon as recall increases, the performance is not notably better for the contrastive

	P (%)	**R (%)**	**F1 (%)**
TF	7.9743	12.6996	9.7969
TF-IDF	**9.8715**	**15.4463**	**12.0451**
PageRank	8.0138	**15.4463**	9.9234
Degree	7.5	12.2707	9.3097
Betweenness	7.6581	12.3733	9.4607
Closeness	7.6185	12.5708	9.4873
Strength	8.0928	13.0759	9.9979
Contrastive betweenness	7.9743	12.8992	9.8557

Table 4.4: Performance of statistics methods and centrality measures in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using exclusively the abstract of 10% research publications of the dataset.

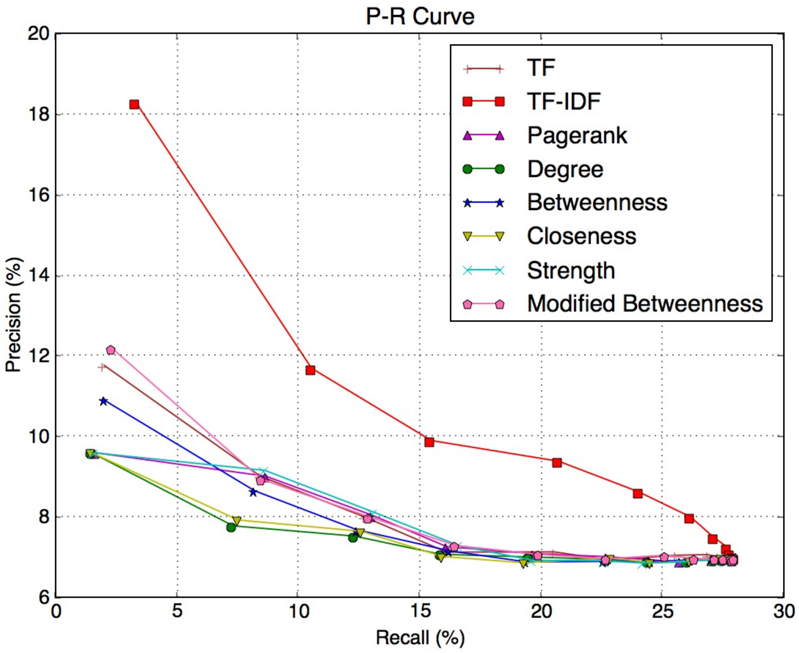


Figure 4.5: Results of the contrastive betweenness (a.k.a. modified betweenness) performance compared to the baseline statistics methods and other centrality measures. This key-phrase extraction experiment was performed using exclusively the abstract of 10% research publications of the dataset.

betweenness compared to other centrality measures. The maximum recall for this experiment is about 27.94%, which is an indicator that many key-phrases are in the body of the paper instead of the abstract.

In Table 4.5 we present a sum of the performance of each statistics method and centrality measure (including the contrastive betweenness) on our dataset when extracting key-phrases with $TF > 1$ from 5% of the papers. The highest score of each evaluation measure has been highlighted. In this condition, we expect statistics methods to present a better performance since they rely on the frequency of the key-phrases and we are filtering out the key-phrases that just appear once. As expected, the best results are obtained using the TF-IDF baseline. Among centrality measures, contrastive betweenness shows the highest performance.

	P (%)	**R (%)**	**F1 (%)**
TF	4.3478	6.2318	5.1220
TF-IDF	**8.3003**	**10.4481**	**9.2512**
PageRank	5.1383	7.1087	5.9650
Degree	5.9288	8.2286	6.8919
Betweenness	5.9288	7.6456	6.6786
Closeness	5.9288	8.2286	6.8919
Strength	3.9525	5.8695	4.7239
Contrastive betweenness	6.3241	8.3702	7.2047

Table 4.5: Performance of statistics methods and centrality measures in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using exclusively key-phrases with $TF > 1$ of 5% research publications of the dataset.

To get a better understanding of the performance for each centrality measure, we report in Figure 4.6 their P-R curves. We observe that the best centrality measure for our dataset is contrastive betweenness, but unfortunately it is not able to outperform TF-IDF for key-phrase extraction. We can appreciate that the contrastive betweenness (a.k.a. modified betweenness) curve behaves better than other centrality measures at low recall; overall, we note the contrastive betweenness curve above the rest of other centrality curves. Note that TF-IDF, betweenness and contrastive betweenness curves start at the same value; since we start drawing the curve at top 1 key-phrases, this is an indicator that these three methods usually select the same key-phrase in first position.

We note that the maximum recall is about 52.83%. It is expected to have a recall value around 50% when we eliminate all the key-phrases with $TF = 1$ as a consequence of Zipf's law.

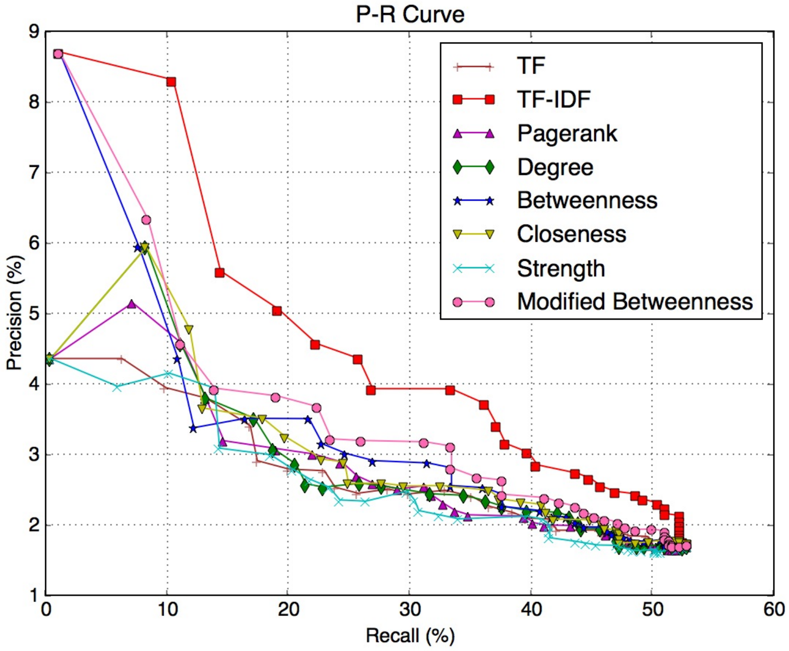


Figure 4.6: Results of the contrastive betweenness (a.k.a. modified betweenness) performance compared to the baseline statistics methods and other centrality measures. This key-phrase extraction experiment was performed using exclusively key-phrases with $TF > 1$ of 5% research publications of the dataset.

Table 4.6 contains the performance of each statistics method and centrality measure (including the contrastive betweenness) on our dataset when extracting key-phrases with $TF = \{1,2\}$ from 5% of the papers. The highest score of each evaluation measure has been highlighted. As we have noticed in the last experiment, the best results are obtained using the TF-IDF baseline. However, this time TF-IDF does not outperform notably the centrality measures, since TF-IDF is an statistic method, it has not the best performance when we are dealing with low frequency key-phrases. The performance of TF-IDF is clearly followed by betweenness and contrastive betweenness.

Our observations are confirmed with the plot in Figure 4.7 where we report the P-R curves. We observe that there is not clear which is the best centrality measure for our dataset and TF-IDF curve is not far from the rest of the centrality curves. We note that the maximum recall is about 27.69%.

	P (%)	R (%)	F1 (%)
TF	0.3952	0.3105	0.3478
TF-IDF	**1.1857**	**1.3457**	**1.2607**
PageRank	0.3952	0.6211	0.4830
Degree	0.7905	0.9316	0.8553
Betweenness	**1.1857**	1.2422	1.2133
Closeness	0	0	-
Strength	0.7905	0.9316	0.8553
Contrastive betweenness	**1.1857**	1.2422	1.2133

Table 4.6: Performance of statistics methods and centrality measures in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using exclusively key-phrases with $TF = \{1,2\}$ of 5% research publications of the dataset.

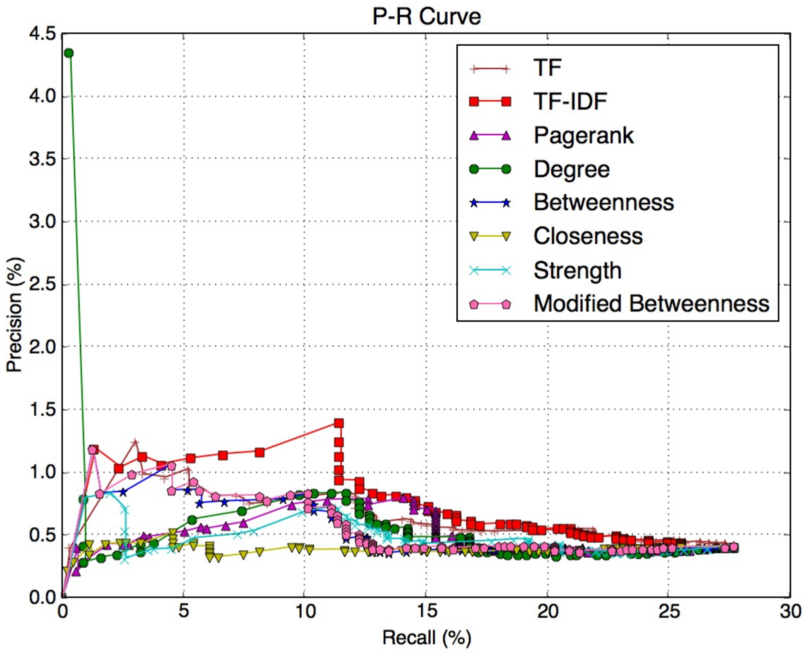


Figure 4.7: Results of the contrastive betweenness (a.k.a. modified betweenness) performance compared to the baseline statistics methods and other centrality measures. This key-phrase extraction experiment was performed using exclusively key-phrases with $TF = \{1,2\}$ of 5% research publications of the dataset.

Summarising the performance of the new contrastive betweenness measure against the other centrality measures and the TF-IDF baseline, we can notice that **contrastive betweenness is able to outperform other centrality measures**, specially when extracting key-phrases with a higher frequency in the document. Unfortunately, its performance is still **far from surpassing the performance that we get with simpler statistics methods such as TF-IDF**.

4.5 Evaluation of weighted sum

To evaluate the results of this approach, we present two stages: (1) training stage, where we used 50% of the dataset to **find an appropriate parameter** α, (2) testing stage, where we used the remaining 50% of the dataset to **evaluate the performance of the weighted sum** with the chosen value of α.

Table 4.7 exhibits the performance of the weighted sum with different values of α parameter when extracting key-phrases from the training set on our dataset. The highest score of each evaluation measure has been highlighted. The best results are obtained using $\alpha = 0.9$, which slightly outperforms the strong TF-IDF baseline. The performance of weighted sum with $\alpha = 0.9$ is clearly followed by $\alpha = 0.75$.

	P (%)	**R (%)**	**F1 (%)**
TF	8.6095	18.7819	11.8068
$\alpha = 0.1$	7.0233	15.5731	9.6807
$\alpha = 0.25$	7.8125	17.2751	10.7592
$\alpha = 0.5$	8.3175	18.2915	11.4352
$\alpha = 0.75$	8.6647	18.8572	11.8736
$\alpha = 0.9$	**8.7042**	**19.0283**	**11.9445**

Table 4.7: Performance of TF-IDF statistics method and weighted sum with different values of α in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using the full text of 50% research publications of the dataset.

To get a broader understanding of the performance for each weighted sum, we report in Figure 4.8 their precision-recall curves for this experiment. In this graph, overall, the best method is weighted sum with $\alpha = 0.75$. For this reason, we decided to choose this as the value of our parameter.

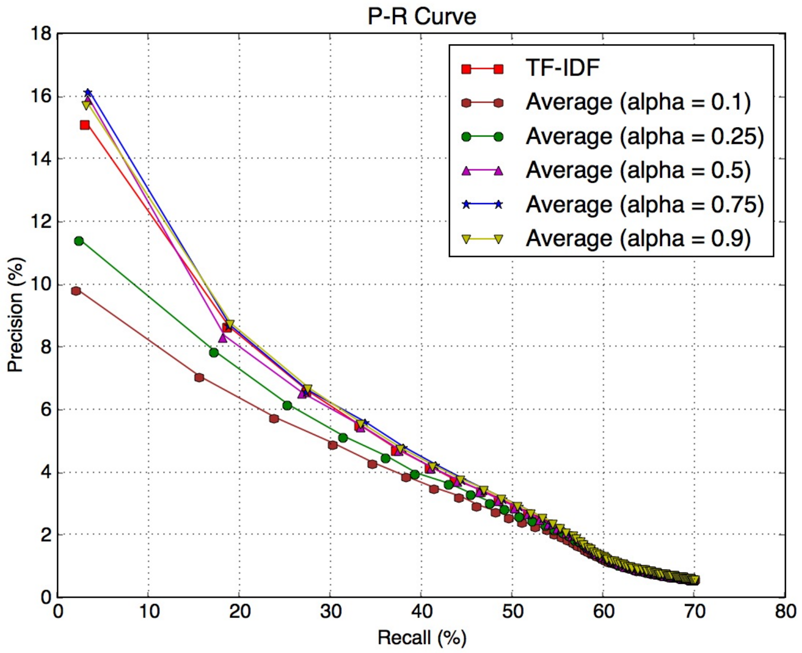


Figure 4.8: Results of the performance of weighted sum with different values of α compared to the TF-IDF baseline. This key-phrase extraction experiment was performed using the full text of 50% research publications of the dataset.

Finally, Table 4.8 presents the performance of the weighted sum with our chosen parameter value $\alpha = 0.75$. In this experiment, it is evident that weighted sum (as expected) outperforms TF-IDF baseline when combining the properties of this strong baseline with the properties of betweenness centrality measure. Figure 4.9 shows the precision-recall curves for this testing stage.

	P (%)	**R (%)**	**F1 (%)**
TF	8.6647	19.4465	11.9880
$\alpha = 0.75$	**8.7279**	**19.6826**	**12.0932**

Table 4.8: Performance of TF-IDF statistics method and weighted sum with $\alpha = 0.75$ in terms of precision, recall and f-score at the top 10 key-phrases on the dataset. This key-phrase extraction experiment was performed using the full text of the remaining 50% research publications of the dataset.

It is expected to outperform the TF-IDF baseline using a weighted sum to combine this score with another centrality measure that includes new properties to re-rank the key-phrases. However, **the computational cost** of improving the score with betweenness centrality as a feature **is not proportional to the amount of improvement** that we can notice in the experiments presented for this approach.

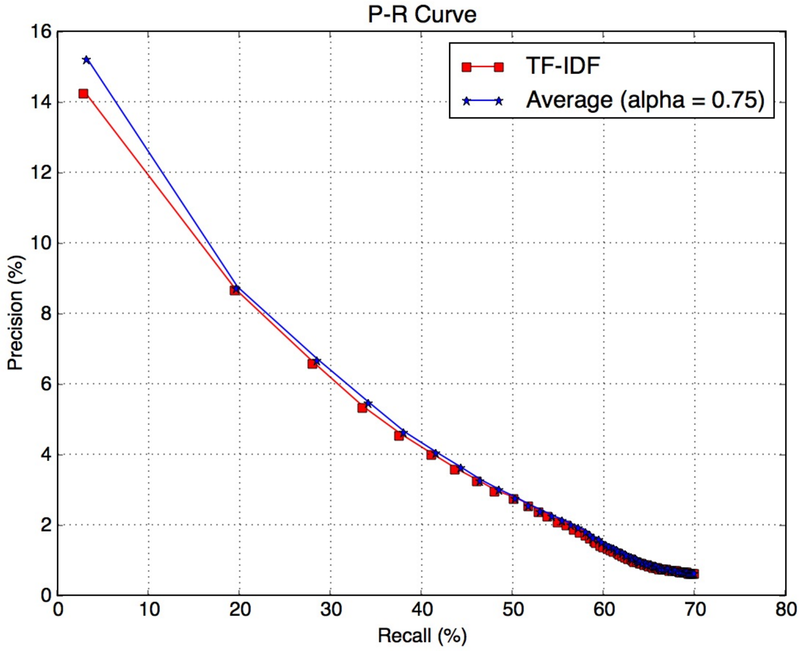


Figure 4.9: Results of the performance of weighted sum with the chosen value of $\alpha = 0.75$ compared to the TF-IDF baseline. This key-phrase extraction experiment was performed using the full text of the remaining 50% research publications of the dataset.

In Table 4.9, we show the top ten key-phrases obtained for document #5, document #10 and document #15 of our dataset, which were ranked according to our main baseline, TF-IDF, and our three approaches, i.e. betweenness, contrastive betweenness and weighted sum with $\alpha = 0.75$. It is difficult to make a qualitative evaluation that indicates if the chosen key-phrases characterise the documents; however, it is interesting to include some examples of the output of each method. For instance, it is easy to note that the three documents have the same top 10 key-phrases according to betweenness and contrastive betweenness. Also, we can note that TF-IDF and the weighted sum have the same key-phrases in their lists, with a slight reordering in the ranking position.

	Expert assigned key-phrases	TF-IDF	Betweenness	Contrastive betweenness	Weighted sum
Document #5	program environ softwar develop environ languag center environ method-bas environ sociolog metaphor structure-ori environ toolkit environ	polici famili model sde individu model infus environ softwar develop environ citi model mechan istar	environ structur tool polici mechan exampl model softwar develop environ program system	environ structur tool polici mechan exampl model softwar develop environ program system	polici environ famili model mechan sde infus softwar develop environ individu model tool structur
Document #10	user interact deleg object-ori design real-tim anim electron book interact illustr	object messag paddl duct control point cach control inquiri herndon control object	object messag system control time interact interact techniqu comput graphic comput exampl	object messag system control time interact interact techniqu comput graphic comput exampl	object messag paddl duct control point control cach control object inquiri herndon
Document #15	tree mathemat multiprocessor interconnect network embed butterfli network complet binari tree wrap-around connect	pe dilat bw signatori pwl string stage embed level binari branch node	embed dilat level node stage pe bw signatori section use	embed dilat level node stage pe bw signatori section use	dilat pe bw embed stage signatori level pwl string node binari branch

Table 4.9: Top ten key-phrases extracted from document #5, document #10 and document #15 of our dataset with the TF-IDF baseline and the three approaches of this work.

4.6 Spearman's rank correlation coefficient

As we have mentioned before, we use Spearman's rank correlation coefficient to get a deeper understanding of the similarities and differences between the unsupervised methods that we have reviewed in this work. We selected Spearman correlation over other correlation coefficients, e.g. Pearson, because we wanted to test for monotonic (but not necessarily linear) relationships. This is, we want to know if two methods preserves the same order of the ranked key-phrases or not.

Spearman's rank correlation coefficient is used when you have two ranked variables, and you want to see whether the two variables covary. If one variable increases and the other variable tends to increase as well, then there is a Spearman correlation close to +1; if the other variable tend to decrease, the Spearman correlation value is close to -1.

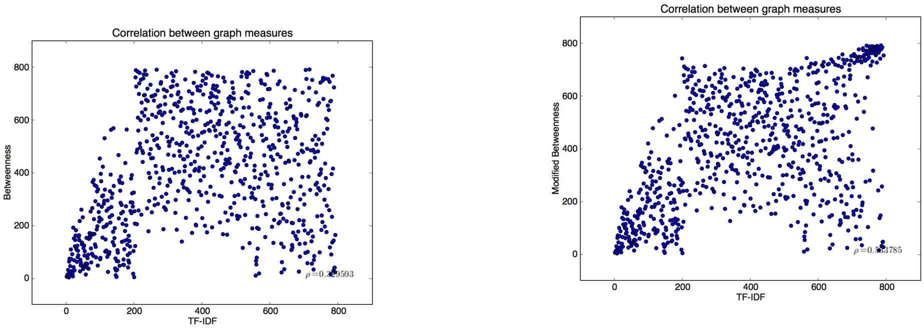
In our case, we used this correlation measure to assess the correlation between the ranking of candidate key-phrases resulting from all the methods that we have presented in this work. Our purpose is to understand whether two methods have the same behaviour or not (if they rank the candidate key-phrases similarly or not). The results are presented in Table 4.10. The interpretation of these results is, for instance: TF and strength are the most correlated methods, and TF-IDF and closeness are the least correlated methods.

Methods	TF	TF-IDF	PageRank	Degree	Betweenness	Closeness	Strength	Modified betweenness
TF	1	0.4952	0.6842	0.7174	0.6398	0.5451	0.9259	0.5896
TF-IDF	0.4952	1	0.3674	0.3811	0.3295	0.2753	0.4881	0.5337
PageRank	0.6842	0.3674	1	0.8463	0.8637	0.3360	0.6821	0.7708
Degree	0.7174	0.3811	0.8463	1	0.8391	0.5153	0.7281	0.7522
Betweenness	0.6398	0.3295	0.8637	0.8391	1	0.4626	0.6440	0.8882
Closeness	0.5451	0.2753	0.3360	0.5153	0.4626	1	0.5555	0.4333
Strength	0.9259	0.4881	0.6821	0.7281	0.6440	0.5555	1	0.5930
Contrastive betweenness	0.5896	0.5337	0.7708	0.7522	0.8882	0.4333	0.5930	1

Table 4.10: Spearman's rank correlation coefficient computed between different key-phrase extraction methods with 5% of the dataset.

As we can observe, with TF-IDF, betweenness is the second method least correlated. For this reason, we decided that the information of betweenness centrality measure is a suitable option to choose as a feature to combine with the TF-IDF in the weighted sum approach.

Additionally, with these results we can notice the improvement that we have had when developing the contrastive betweenness centrality measure. Note that contrastive betweenness is the most correlated method with TF-IDF, which means, **the ranking of candidate key-phrases obtained with TF-IDF presents a higher similarity to the ranking obtained by contrastive betweenness than any other method**. This is desirable if we pretend to outperform (or at least to emulate) the performance of TF-IDF with a graph-based method to find the characteristic key-phrases of a document. Figure 4.10 shows an example to compare the correlation of the key-phrase ranking for: betweenness with TF-IDF (4.10a); and contrastive betweenness with TF-IDF (4.10b). Ideally, if we wanted to obtain the same key-phrase ranking of TF-IDF, we would observe a perfect diagonal in the images. Note that the scatter points in Figure 4.10b looks closer to the diagonal than the scatter points in Figure 4.10a, this is because the Spearman's rank correlation coefficient in 4.10b is higher.



(a) Correlation between TF-IDF and betweenness centrality, $\rho = 0.3295$.

(b) Correlation between TF-IDF and contrastive betweenness (a.k.a. modified betweenness), $\rho = 0.5337$.

Figure 4.10: Visualisation of the correlation between the baseline method TF-IDF and our approaches.

Finally, find in Figure 4.11 the visualisation of the correlation between the key-phrase ranking of betweenness versus contrastive betweenness. As we can note, even when contrastive betweenness is having a different behaviour than simple betweenness centrality, the correlation coefficient is still high, therefore, it is an explanation of why contrastive betweenness is not outperforming TF-IDF baseline: **the ranking of candidate key-phrases obtained with contrastive betweenness is still very similar to that obtained with simple betweenness centrality**.

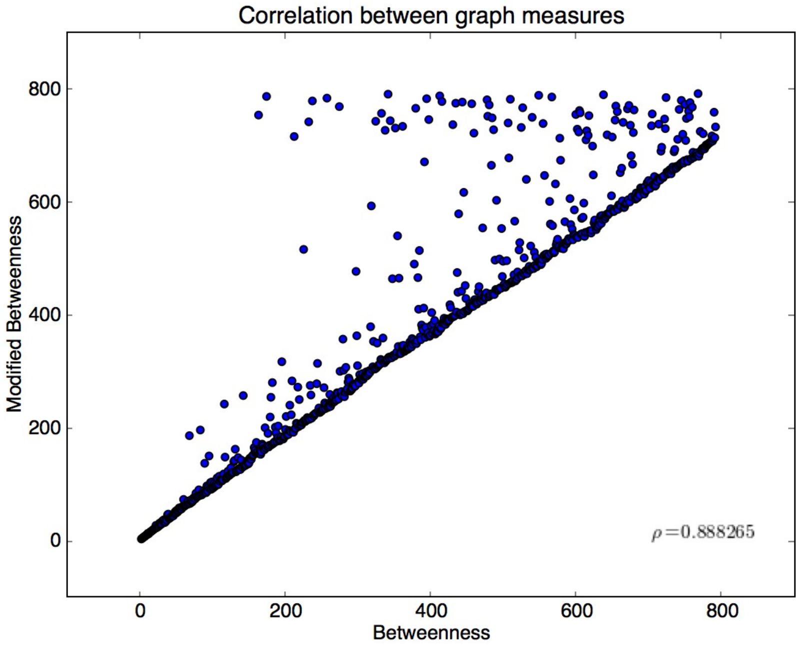


Figure 4.11: Visualisation of the correlation between our first approach, betweenness centrality, and our second approach, contrastive betweenness (a.k.a. modified betweenness), $\rho = 0.8882$.

Chapter 5

Conclusion

In this work we presented a study on key-phrase extraction methods using statistics measures, centrality measures and supervised feature combination (weighted sum). We designed a new method, i.e. contrastive betweenness, based on centrality concepts to intend to get the key-phrases that characterise a research paper.

We tested our approaches on a real dataset developed by Krapivin et al. (2008) that consists of 2304 papers in the field of Computer Science.

We found that betweenness, closeness and contrastive betweenness were among the top performer centrality measures, while TF-IDF statistics was always the best performer in the task of key-phrase extraction.

Our results show a potential ability of producing a high quality extraction of key-phrases with graph-based unsupervised systems , which have the main advantage to adapt to real-time tasks without the need of a training stage or labelled data.

Some specific observations are presented below:

TF-IDF: This is a well-known score for ranking the importance of key-phrases in a text. This is the baseline of this work and was outperformed only by the weighted sum approach. **TF-IDF showed a fair performance in key-phrase extraction** from the full text of a document and from the abstracts. **TF-IDF only performed poorly when extracting key-phrases with a low TF**. This is expected because TF-IDF has the disadvantage of relying in the frequency of the terms. Although TF-IDF formula is easy and fast to compute for the entire corpus, this method hardly can be applied for real-time tasks where the documents appear once at a time, since we need a complete corpus to reliably estimate inverse document frequency (IDF).

Graph construction: We built co-occurrence networks where nodes are represented as noun phrases extracted from the preprocessed text and the edges are links between noun phrases that co-occur within a window of each other; we set the window size to ten. It is relatively easy to get the graphical representation of a document in this way. Although **graphical representations are able to keep the structure of the document**, we have to select carefully the conditions to build the graph. Co-occurrence networks does not consider the semantics of a sentence, which is a disadvantage compared to other networks that include linguistic considerations to build the graph. There are plenty ways to experiment the construction of a graph and there is not standardisation. We also recommend to **keep more information of the document in the graph with weighted edges to indicate the degree of node relationships**, e.g. the co-occurrence frequency of a pair of phrases, the distance of two phrases in a text, etc.

Centrality measures: To rank nodes in the co-occurrence graphs we used five centrality measures (degree, strength, PageRank, closeness and betweenness). As expected, some centrality measures perform better under different conditions. **Betweenness is a good measure when we extract key-phrases from the full text of a document** and **closeness performs better when we use only the abstract section to extract key-phrases**. We conclude that **centrality measures can be used for key-phrase extraction without the need for a large external database** to reliably compute the IDF component of TF-IDF. Furthermore, with the computation of Spearman's rank correlation coefficient, we concluded that some of these measures yield almost identical rankings. For example PageRank and degree, therefore, depending on the corpus and the type of key-phrases that we want to extract, we can use simpler centrality measures such as degree instead of more complex and more costly measures, e.g. PageRank. Finally, **graph-based methods have the great advantage of being highly portable to other domains and languages**.

Contrastive betweenness: We designed an extension to betweenness centrality in an intend not just to extract important key-phrases, but to find the characteristic key-phrases of a research paper. This algorithm relies on the property of betweenness centrality to measure how easy is to get to the rest of the paper from a key-phrase, but our intuition was that we could add the way to measure how difficult is to escape the document from a key-phrase by subtracting the betweenness centrality obtained in a huge graph that comprises all the papers of the corpus. This new measure **overall performed better than any of the other five centrality measures**; but, unfortunately, the results **did not outperformed the TF-IDF baseline**. Besides, the computational cost of computing this new measure is high compared to the other graph-based methods. However, with the design and implementation of contrastive betweenness, we have shown that **there is still an open and unexplored research field in graph algorithms**, whose concepts need to be studied and extended in order to capture the complexity of key-phrase extraction task.

Weighted sum: This method is an hybrid system that takes advantage of the strengths of the frequency-based methods and the graph-based methods to combine their features and obtain a better reordering. We included TF-IDF and betweenness scores in the weighted sum because the correlation study indicates that they are least correlated, therefore, we eliminate redundancy by selecting this measures as our features. The results show a **slightly better performance compared to the simple TF-IDF baseline**, which indicates that this can be a direction of study to improve key-phrase extraction. However, a deeper study should be made to select optimal features for the weighted sum method.

Evaluation: We used a real dataset of 2304 papers in the field of Computer Science with gold standard key-phrases annotated by experts in the domain. To evaluate our rankings we used three popular metrics (precision, recall and f-score). These metrics are a reference to compare our results with previous work and prove that we are obtaining results in the same range of precision, recall and f-score as other state-of-the-art work. Nevertheless, we have still not developed a metric that is able to **measure the uniqueness of a key-phrase,** this is, if this characterises a document or not.

At last, we hope this work contributes to the field with a comprehensive study to encourage future research using graph-based methods and developing variations of centrality measures to improve unsupervised key-phrase extraction.

5.1 Future work

As we have mentioned, key-phrase extraction task is far from being solved, as reflected by the poor state-of-the-art results on different datasets. Our analysis revealed that there is much work ahead and we would like to highlight some open questions:

- Interesting directions for future work consist of exploring different types of networks, e.g. varying the window size, using POS tags to link nodes, using directed edges.
- Further research is expected for re-ranking with weighted sum method, where a possible direction is to design a proper experiment to select the most appropriate centrality measures as features to combine in supervised key-phrase extraction.
- In the future work, it will also be necessary to extend these key-phrase extraction systems considering texts of different domains, other languages (different than English) and other datasets to confirm that graph-based methods are easily transferable.
- Future work on automatic key-phrase assignment algorithms will focus not only on extracting key-phrases that occur in the document, but in **generating** new, descriptive key-phrases that does not occur in the text, a key weakness as discussed in our key-phrase analysis, where we found that 25% of the expert assigned key-phrases does not occur in the documents.
- Finally, it may be possible to design better methods to extract characteristic key-phrases more accurately from documents, as we intended in this work; nevertheless, it is important to consider that if these systems include sophisticated features, such as contrastive betweenness, we would have to develop faster algorithms to manage scalability issues when handling long documents and big datasets.

Bibliography

Abilhoa, W. D. and de Castro, L. N. (2014). A keyword extraction method from twitter messages represented as graphs. *Applied Mathematics and Computation*, 240:308–325.

Beliga, S., Meštrović, A., and Martinčcić-Ipšić, S. (2014). Toward selectivity based keyword extraction for croatian news. *arXiv preprint arXiv:1407.4723*.

Boudin, F. (2013). A comparison of centrality measures for graph-based keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 834–838.

Brandes, U. (2001). A faster algorithm for betweenness centrality*. *Journal of Mathematical Sociology*, 25(2):163–177.

Chen, P.-I. and Lin, S.-J. (2010). Automatic keyword prediction using google similarity distance. *Expert Systems with Applications*, 37(3):1928–1938.

Ferret, O. (2002). Using collocations for topic segmentation and link detection. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction.

HaCohen-Kerner, Y. (2003). Automatic extraction of keywords from abstracts. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 843–849. Springer.

Haddoud, M., Mokhtari, A., Lecroq, T., and Abdeddaïm, S. (2015). Accurate keyphrase extraction from scientific papers by mining linguistic information. In *Proc. of the Workshop Mining Scientific Papers: Computational Linguistics and*

Bibliometrics, 15th International Society of Scientometrics and Informetrics Conference (ISSI), Istanbul, Turkey: http://ceur-ws. org.

Hasan, K. S. and Ng, V. (2010). 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.

Hasan, K. S. and Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. *Proceedings of the Association for Computational Linguistics (ACL), Baltimore, Maryland: Association for Computational Linguistics.*

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics.

Hulth, A., Karlgren, J., Jonsson, A., Boström, H., and Asker, L. (2001). Automatic keyword extraction using domain knowledge. In *Computational Linguistics and Intelligent Text Processing*, pages 472–482. Springer.

Jiang, X., Hu, Y., and Li, H. (2009). A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 756–757. ACM.

Kim, S. N. and Kan, M.-Y. (2009). Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*, pages 9–16. Association for Computational Linguistics.

Krapivin, M., Autayeu, A., and Marchese, M. (2008). Large dataset for keyphrases extraction. Technical Report DISI-09-055, DISI, Trento, Italy. http://eprints.biblio.unitn.it/archive/00001671/01/disi09055-krapivin-autayeu-marchese.pdf.

Krapivin, M., Autayeu, A., Marchese, M., Blanzieri, E., and Segata, N. (2010). Keyphrases extraction from scientific documents: Improving machine learning approaches with natural language processing. In Chowdhury, G., Koo, C., and Hunter, J., editors, *The Role of Digital Libraries in a Time of Global Change*, volume 6102 of *Lecture Notes in Computer Science*, pages 102–111. Springer Berlin Heidelberg.

Lahiri, S., Choudhury, S. R., and Caragea, C. (2014). Keyword and keyphrase extraction using centrality measures on collocation networks. *arXiv preprint arXiv:1401.6571*.

Leydesdorff, L. (2007). Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. *Journal of the American Society for Information Science and Technology*, 58(9):1303–1319.

Li, W. (1992). Random texts exhibit zipf's-law-like word frequency distribution. *Information Theory, IEEE Transactions on*, 38(6):1842–1845.

Litvak, M. and Last, M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24. Association for Computational Linguistics.

Liu, Z., Chen, X., Zheng, Y., and Sun, M. (2011). Automatic keyphrase extraction by bridging vocabulary gap. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 135–144. Association for Computational Linguistics.

Lopez, P. and Romary, L. (2010). Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 248–251. Association for Computational Linguistics.

Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Matsuo, Y. and Ishizuka, M. (2002). Keyword extraction from a document using word co-occurrence statistical information. *Transactions of the Japanese Society for Artificial Intelligence*, 17:217–223.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. Association for Computational Linguistics.

Nguyen, T. D. and Kan, M.-Y. (2007). Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326. Springer.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.

Romero, M., Moreo, A., Castro, J. L., and Zurita, J. M. (2012). Using wikipedia concepts and frequency in language to extract key terms from support documents. *Expert Systems with Applications*, 39(18):13480–13491.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sonawane, S. and Kulkarni, P. (2014). Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96(19).

Stephenson, K. and Zelen, M. (1989). Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1–37.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

Wan, X. and Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.

Wartena, C., Brussee, R., and Slakhorst, W. (2010). Keyword extraction using word co-occurrence. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, pages 54–58. IEEE.

Xie, Z. (2005). Centrality measures in text mining: prediction of noun phrases that appear in abstracts. In *Proceedings of the ACL Student Research Workshop*, pages 103–108. Association for Computational Linguistics.

Yih, W.-t., Goodman, J., and Carvalho, V. R. (2006). Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, pages 213–222. ACM.

Zhou, Z., Zou, X., Lv, X., and Hu, J. (2013). Research on weighted complex network based keywords extraction. In *Chinese Lexical Semantics*, pages 442–452. Springer.