NTNU

Norwegian University of
Science and Technology

Graph representation of documents content and its suitability for text mining tasks

Adrian Viaño Iglesias

Master of Science in Computer Science
Submission date: June 2011
Supervisor: Kjetil Nørvåg, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

The goal of this project is to investigate how to represent the content of document collection in a form of a graph and whether we can use graph algorithms to realize traditional text mining tasks as text clustering and classification. In recent years, methods able to extract key concepts and their relations from the text emerged. This enables the representation of documents content in a form of a graph. Possible methods usable for this task include association rules networks, wikification methods (annotation of text with Wikipedia concepts), named entity recognition methods. Complex networks and graphs are prominent research topic nowadays, due to the availability of large scale real-world data in graph form (e.g. social networks, information networks, communication networks, biological nets) and there has been a good progress in algorithms for analyzing large scale graph. In this project we will combine those two branches of research and use graph algorithms (such as community detection, i.e., identification of densely connected clusters of nodes) on the networks build from the content of document collection(s). We will investigate the suitability of such approach to the text mining task.

Assignment given: 15. January 2011
Supervisor: Kjetil Nørvåg, IDI
Co-supervisor: Marek Ciglan, IDI

Abstract

Association rules mining is one of the the most relevant techniques of data mining. It has been also applied in the domain of text mining, but the results are hard to interpret. In this matter, an Association Network is an structure to represent as a graph the relationships mined as association rules. The goal of this project was to provide a methodology to build association networks from concepts extracted from a collection of documents, as well as the study of the mathematical properties of the association networks to prove that they are not random graphs and that they exhibit small-world properties.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Description

Association rules mining is one of the best known data mining algorithms. It has been successfully used in numerous application domains from the marked basket analysis to intrusion detection. Researchers have tried to use association rules mining also in the domain of text mining, although the results are often hard to interpret. The main reason is the sheer amount of the rules produced by mining algorithms. As the user is usually intended to be the end consumer of the mined rules (e.g. domain expert examining the rules and applying it to his problem domain) the large amount of the generated rules is a serious drawback.

The aim of our work is to investigate when can we exploit the large amount of the produced rules to our advantage and benefit from it rather than consider it to be a drawback. The main idea is to interpret a rule $A \rightarrow B$ as a relationship between item A and B. We interpret this rule as a directed edge connecting the items A and B. In this way, we can construct a network where the nodes correspond to the items of our dataset and edges represents the relations mined as association rules. We will refer to such a network as an association network. Our main research question is what are the mathematical properties of association networks. If the properties resembles those of random graphs, the association networks would not be very valuable. On the other hand, if association networks exhibits some non-random properties, we could exploit them to gain more insight on the hidden structure of the natural language texts. Most of the real-world networks have scale-free and small-world properties. If this would be true also for association networks, implications might be interesting. In that case, we could use network link analysis algorithms and graph mining techniques to analyze unstructured text collections. This would allow us to bridge the text mining and graph mining domains to certain extend.

The main goal of this work is to provide methodology to build association networks from the concepts extracted from natural language documents and to provide an extensive study of association networks properties. We study the effects of different concept extraction methods and different interestingness measures of association rules on the structure of the generated networks.

1.2 Motivation

In this project we describe a solution to construct a new structure denominated association network based on a set of association rules extracted from a collection of documents. The solution will be obtained using some existing concept extraction methods (e.g. Maui Indexer, Wikipedia Miner and Topia Term Extract) to get the keywords of each document of the collection. We want to study if the concept extraction method chosen has relevance in the properties of the final association network. With the obtained terms we will extract the association rules according to different interestingness measures. The aim, as well as with the extraction methods, is to study if the properties of the association network vary depending on the measure used. Given a concept extraction method and a interestingness measure, the properties of the resultant association network will be studied, specifically we will compare the clustering coefficient, the average path length and the degree distribution with the ones corresponding to a random graph. Once we verify these properties of the association networks the implications are very interesting because this will allow us to use the network in the real world. The goal of the second part of our project is to use this association network in some practical applications that are already carried out using classical techniques. We propose some of those possible uses and we study with more details one of them, specifically topic modeling. We will apply a community based measure to obtain the more relevant words of a document and the results will be compared with the other existing methods.

1.3 Outline

Here is a description of the chapters in this document.

Related Work: The first chapter introduces an overview of previous works that have some relation with this project. Most of these researches belong to the field of data mining that represents an important part of this project. More specifically text mining and association rule mining.

Association Networks Construction: The main part of this master thesis is the construction and study of the properties of the association rules networks. This section explain the different phases to construct an association network and the following evaluation of its properties.

Applications of Association Networks: Once the association network is constructed, in this section some possible applications are proposed and one experiment with one of those applications is detailed and compared with some other classical methods.

Chapter 2

Background

In this chapter, in order to give a wider vision of the domain, we explain some previous existing theories in which our project is based upon and some topics that are just relevant to understand our research. Then we will give an introduction to some data mining techniques on which it is based a big part of our work. At the end of this chapter we will describe some classical techniques of topic modeling. When we study the possible applications of the association networks, the results obtained with the classical methods will be compared with the ones obtained with the association network.

2.1 Preliminaries

In this section we will introduce some preliminary concepts such as general definitions and notations, most of them related with the graph theory.

2.1.1 Graphs

A graph is a pair $G = (V, E)$. The elements of V are the vertices (or nodes) of the graph G and the elements of E are its edges. Usually a graph is pictured by drawing a dot for each node and joining two of these dots by a line if they are connected by an edge. An edge can be either directed or undirected. If the graph is directed, the edges are drawn as arrows. There are other ways of visualizing and representing graphs that sometimes are also useful. For example, the intersection graph or the interval graph.

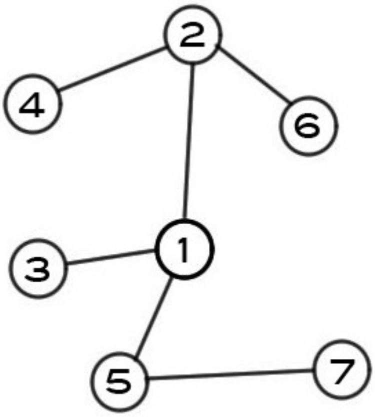

Figure 2.1: Example of undirected graph with seven nodes

Properties

Graphs have some properties that must be taken into account. The *order* of a graph is the number of vertices. The *size* of a graph is the number of edges. The number of edges connected to a node is known as the *degree* of the node. The *distance* between two vertices in a graph is the number of edges in a shortest path connecting them. The *connectivity* of a graph is an important measure of its robustness as a network. The connectivity is the size of a smallest vertex cut that is a partition of the vertices of a graph into two disjoint subsets. Two edges are *adjacent* if they have a common vertex and two vertices are adjacent if an edge joining them. An edge is *incident* to a vertex if it joins it to another. *Weighting* corresponds to a function that to each edge is associated with a value (cost, weight, length, etc..), to increase the expressiveness of the model. This is widely used for optimization problems, such as the traveling salesman or the shortest route. *Labelling* is a distinction made to the vertices and edges with a brand that makes them uniquely distinguishable from the rest.

2.1.2 Small-World Networks

In the 1960's the psychologist Stanley Milgram began an experiment called as Small World phenomenon in Harvard University, concluding that could be connected to two people in the U.S. with only six jumps on average, this phenomenon is called: six degrees of separation. In this experiment he began the investigation of a particular category of small world networks. In 1998, mathematicians Duncan Watts and Steven Strogatz conducted a study focused on network analysis focused on some kind of random graphs showing peculiar connectivity properties. This mathematical graph is the *Small-World Network* in which most nodes are not neighbors of one another, but most nodes can be reached from every other by a small number of hops or steps.

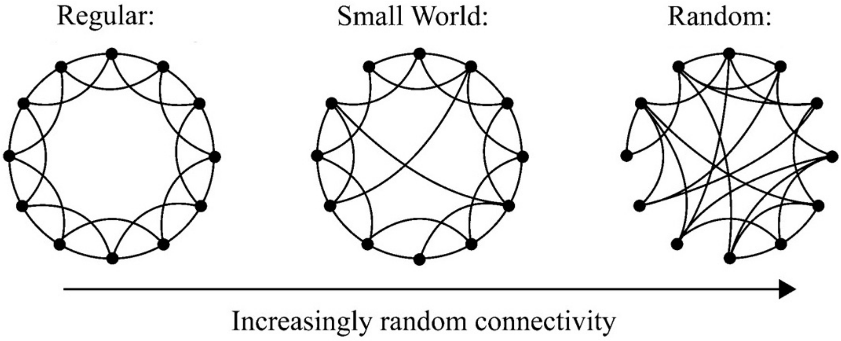

Figure 2.2: Simple model of networks with regular short-range bonds and random long-range bond [35]

Some of the properties of the small-world networks are:

- **Average Path Length:** The shortest path length of a graph is the way with the minimum number of steps from one node to any other in the graph. If the value of the average path length is small, in general only some step are needed to reach the other node. Consider a graph G with n nodes then the average path length is given as:

$$L = \frac{1}{(n)(n-1)} \sum_{i,j \in V(G)} d_{ij}$$

where $i \neq j$ and d_{ij} represents the minimum number of edges required to connect i with j

- **Clustering Coefficient:** This property measures the probability that given three different nodes and two of them are neighbors, the third node is also a neighbor. It is given as:

$$C = \frac{1}{n}\sum_{i=1}^{n} C_i$$

where C_i denotes the clustering coefficient of the subgraph G_i of the graph G.

- **Efficiency:** This property represent how the graph transmits the information through the nodes. inversely proportional to shortest distance between two vertices.

$$e_{ij} = \frac{1}{d_{ij}}$$

$$E(G) = \frac{1}{(n)(n-1)} \sum_{i,j \in V(G_i)} e_{ij}$$

It can be measured locally or globally.

A *clique* in an undirected graph G is a set of nodes N that for every pair of nodes in N, exists an edge connecting these two nodes. Alternatively, a clique is a graph in which every vertex is connected to every other vertex in the graph. This is equivalent to say that the subgraph induced by N is a complete graph. The size of a clique is the number of nodes that contains the clique.

Other property of the small-world networks is that they have a degree distribution that can fit with a power law distribution. A new model called scale-free network was proposed in [3, 4] to explain the power law distribution.

A Scale-Free Network is a network that follows a power law degree distribution. The power law has very surprising mathematical properties and they appear in many situations of real life, for example the population of cities or the intensities of earthquakes. Mathematically, a scalar x follow a power law distribution if it is extracted from a probability distribution:

$$p(x) = x^{-\alpha}$$

where α is a constant that is known as *scaling parameter* and it usually has values between $2 \lesssim \alpha \lesssim 3$. These are just approximated values and some exceptions can be found. In order to determine if a distribution fits with the power law distribution, we have to look its behavior in a plot. In order to decide if a particular network has the properties of Small-World networks or Scale-Free networks, its characteristics (clustering coefficient, average path length and degree distribution) have to be compared with the corresponding in a random graph.

2.2 Data Mining

Data mining is a research field that study how to extract some patterns from large data sets. Data mining is a very general field and it involves some more specific techniques. In this section we introduce those techniques that are more related with our project.

2.2.1 Text Mining

A huge number of text documents started to appear on Internet since the beginning of its creation. The web, databases and many other online sources began to be seen as a huge repository of online information and many researchers and scientists focused their work discover knowledge from these textual databases. This research field is known as Text Mining.

Text mining is nearly as old as information retrieval (IR) itself and it is a special kind of data mining. Text mining represents an important and difficult challenge mainly due to the properties of the natural language used in most of the available documents. Nevertheless, text is the most used way for the formal exchange of information. Due to this, the objective of text mining is to reduce the effort required to obtain useful information from large text collections. For example, text mining can be used to analyze messages and emails to try to filter spam according to some patterns or simply to analyze all the received emails and automatically send them to the right department or employer.

Preprocessing

Usually in data mining the data is structured and all the available tools are prepared to obtain the knowledge easily. But in text mining, the information is unstructured and even very ambiguous. A manual analysis of the text documents is not possible and not very effective. The use of automatic tools for analyze the large amount of documents is highly recommended. To avoid the problem of the unstructured information, the first step in every text mining methodology should be a preprocessing phase to prepare the text for the analysis. A morphological analysis is a good point to start. Documents are filtered to find the smallest unit of the document, after this, the document is tokenized to explore the words in a sentence to discard and remove the unimportant words. Normally the words ignored are articles, prepositions, conjunctions, pronouns, adverbs and non-informative verbs (e.g. be, have). In addition, special characters, parentheses, commas, etc., can be replaced with an space between words in the documents.

The next step is called stemming and is a technique to reduce the words into their root. Many words in the natural language appears in different composed forms (e.g. agreed, agreeing, disagree, agreement and disagreement). All these words can be transformed into the original or base form, in this example they belong to the word agree. Nevertheless, the final result of the reduction of the words may cause that some word are transformed into a wrong root or stem. However, these words do not have to be a problem for the stemming process if they are not used for human interaction. The stem is still useful, because all other endings of the root are transformed into the same stem. Some systems could have problems if they are case sensitive and compare a word in capital letter with the same word in lower case. In the english language most of the endings are simple and similar, this property make easier the existence of algorithms to do the stemming. Some other languages such as french, spanish or portuguese have endings that can change the root of a word. In this case, the brute force method is a good option. There are many other algorithms to perform the stemming like Lovins stemming, Porter stemming, Krovetz stemming, Inflectional stemming, derivational stemming. [8] Remove the prefix of a term could be dangerous. The reason of this is that the meaning of a word can be completely different if the prefix is deleted because, for example, there are some prefixes that are used to obtain the antonym of a word. Different methods to perform the stemming can be found. One simple way to do it is the brute force method. This requires a dictionary which contains the endings of a word. This method has some serious disadvantages like that the process, to store all the information will spend many of the available resources. The other disadvantage is that the dictionary usually does not contain all the endings for each root.

After the morphological analysis, is time to start to analyze the syntax. Information retrieval systems will get better results if the syntax of the sentences are studied. The recognition and identification of elements in a sentence like nouns, verbs, adjectives or prepositions is called part-of-speech tagging (POS tagging) [7]. Some parts are more important than others in the text. Nouns are the most important words in a sentence or document. If in some documents appear many times the same term, is supposed that both also have a similar topic. A part-of-speech tagger can be implemented in many different ways like based on dictionaries, rule-based methods or statistical methods. A general tagger for all the languages cannot be implemented because languages are completely different each other. Some research studies have prove that the use of statistical methods have better results than rule-based methods. Probably the best results were obtained with the probabilistic Hidden Markov model [39].

Information about the topic of a document is provided by the number of occurrence of a term. There are many approaches to obtain the main topic, for example simply calculate the term frequency of each word in the text. Nevertheless the most commonly used approach is the term frequency - inverse document frequency (tf-idf) model [25]. In this model, having a collection of documents, the importance of a word in a particular document increases with the number of times that this word appears in this document but is offset with the number of times that appear in all the documents of the collection.

$$tfidf(w) = tf * \log\left(\frac{N}{df(w)}\right)$$

tf = number of word occurrences in a document.
df = number of documents containing the word.
N = number of all documents.

In other words, if a term appears many times in a document but also appears many times in the collection it will get a lower score. This is very interesting in order to penalize common words that appear very often in all documents and which are not part of the main topic.

2.2.2 Concept Extraction

A concept is an idea or topic present in a document. A concept have more concrete meaning than a simple keyword because one concept can represent to several terms in the text. Concept extraction is the technique that given a collection of documents in a domain, discover the set of concepts that describes a particular document. It is very important that the all the documents in the collection have similar topic in order to discriminate the unimportant words unrelated with the real topic. Due to the natural language is ambiguous this also a difficult task. Many words present in a text are not important and lists with stop words are used to discriminate them. A main property of the concept extractors is most of them not only get simple words as concepts, they also can extract composed expressions such as ”Barack Obama”. Concept extractors need a collection of documents with its concepts already extracted to train and learn how to extract the concepts for all documents. In a document, usually those words that appears frequently in the text are more important to represent the content of the document than the terms that only appear few times. Some methods [38] increase the value of a term that appears many times as itself or with morphological derivations. The value of a term (weight) can be increased more or less depending in which section of the document is found. A term will be more important if it appears in the title or the abstract than in the corpus. The terms with a weight higher than a predefined threshold are selected to represent the concept. The term with the highest weight will represent the topic concept.

Maui Indexer, Wikipedia Miner and Topia Term Extract are three examples of concepts extraction methods.

2.2.2.1 Maui Indexer

Maui Indexer has been developed by Olena Medelyan as a part of her PhD project. Maui builds on the keyphrase extraction algorithm Kea and apart of other features it also allows the assignment of topics to documents based on terms from Wikipedia using Wikipedia Miner.

Topic indexing can be realized in three different ways:

1. Term assignment (also referred to as keyphrase indexing and subject

indexing) uses domain-specific controlled vocabulary as a source of topics.

2. Keyphrase extraction (also called keyword extraction) identifies phrases that are prominent in a given text. However, it can be made more consistent if topics are chosen with respect to Wikipedia.

3. Tagging is an example of keyphrase extraction, where tags are freely chosen and are not restricted to Wikipedia.

In order to generate and filter the candidate Maui Indexer implements a two-stage algorithm for performing these tasks automatically. The first phase, candidate generation, identifies candidate topics in a given document. Candidates are either mappings from phrases to terms in the vocabulary (dictionary in term assignment or Wikipedia in controlled keyphrase extraction), or document phrases (in tagging). The second phase, filtering, analyzes the properties, or features, of the candidate topics and filters out the most significant ones. Maui utilizes several kinds of features:

- Frequency statistics, such as term frequency, inverse document frequency, TF-IDF;
- Occurrence positions in the document text, e.g. beginning and end, dispersal of occurrences;
- Keyphraseness, computed based on topics assigned previously in the training data, or particular behavior of terms in Wikipedia corpus;
- Semantic relatedness, computed using semantic relations encoded in provided dictionary, if applicable, or using statistics from the Wikipedia corpus;

Maui uses machine learning to capture the typical feature values of topics assigned manually to training documents and then applies the generated model to assign topics to unseen documents.

2.2.2.2 Wikipedia Miner

The online encyclopedia Wikipedia is a giant multilingual database of concepts and semantic relations. This collection has been used by researchers and developers in their own projects. The Wikipedia Miner toolkit provides an object-oriented method to access to Wikipedia's structure and content. The Wikipedia Miner collection is structured in classes. All Wikipedia's content is presented on pages. Each *page* contains one article that is the content and also contains *redirects* that connect the article with alternative titles, *categories* related with the article to link the article with more general topics. The Wikipedia Miner toolkit includes algorithms for generating semantic relatedness measures, which measure the extent to which different words or concepts relate to each other. These measures have a wide range of applications because they allow terms and concepts to be compared, organized, and reasoned with. Wikipedia Miners relatedness measures are generated using the hyperlinks made between Wikipedias articles. These articles reference each other extensively, and at first look the links appear to be promising semantic relations but also contains links to many irrelevant concepts.

2.2.2.3 Topia Term Extract

Given a piece of content, this extraction method determines the important terms by making use of a simple Parts-Of-Speech (POS) tagging algorithm. POS Taggers use a lexicon to mark words with a tag. Since words can have multiple tags, the determination of the correct tag is not always simple. The first step of tagging is to tokenize the text into terms. While most tokenizers ignore punctuation, it is important for Topia to keep it, since it needs it later for the term extraction. The next step is tagging that is done in two phases. During the first phase terms are assigned a tag by looking at the lexicon and the normalized form is set to the term itself. In the second phase, a set of rules is applied to each tagged term and the tagging and normalization is tweaked. Once the text is tagged, it is time to start to look at the term extractions. If the term consists of a single word, at least 3 occurrences of a term must be detected to take that term in consideration. Multiword nouns and proper names occur only once or twice in a text but they are very important terms. To handle this, the concept of strength was introduced. Terms with strength higher than 1 are selected even if they only appear once in the text.

2.2.3 Association Rule Mining

An association rule is defined in the following way in [1]: Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subset I$. An association rule is an implication of the form $X \rightarrow Y$ where $X \subset I, Y \subset I$, and $X \cap Y = \phi$. The rule $X \rightarrow Y$ holds for the dataset D with support s and confidence c if $s\%$ of transactions in D contain $X \cup Y$ and $c\%$ of transactions in D that contain X also contain Y. Support is the percentage of transactions containing an item set, calculated in a statistical manner, while confidence measures the strength of the rule.

Given a support of an item set denoted by $supp(I_k)$, and the rule $I_1 \rightarrow I_2$, the support and the confidence of the rule denoted by $Supp(I_1 \rightarrow I_2)$ and $Conf(I_1 \rightarrow I_2)$, respectively, are calculated as follows:

$$\begin{aligned} Supp(I_1 \rightarrow I_2) &= supp(I_1 \cup I_2) \\ Conf(I_1 \rightarrow I_2) &= \frac{supp(I_1 \cup I_2)}{supp(I_1)} \end{aligned}$$

The constrains of minimum support and minimum confidence are established by the user with two threshold values, one for the support and other one for the confidence. A strong rule is an association rule whose support and confidence are greater that the thresholds. Support and confidence were the original measures proposed for association rules [1]. Nevertheless many other measures can be used to extract the association rules as we will see in the next section where we show some examples of interestingness measures. Once the user has determined these values, the process of obtaining association rules can be decomposed in two different steps:

- Find all the itemsets that have a support higher than the threshold. These itemsets are called frequent itemsets.
- Generate the association rules based on the frequent itemsets.

Association rule mining usually find many rules. An algorithm finds all rules that satisfy support and confidence requirements. Different algorithms will find the same results. Next to confidence some other interestingness measures for rules were proposed, the filter of the rules will be different depending on the measure used.

Interestingness Measures

In the table 2.1 there are some common objective interestingness measures for association rules. In this table A and B represent the antecedent and the consequent of a rule, respectively.

Measure	Formula
Support	$P(AB)$
Confidende	$P(B\|A)$
Coverage	$P(A)$
Prevalence	$P(B)$
Recall	$P(A\|B)$
Specificity	$P(\neg B\|\neg A)$
Accuracy	$P(AB) + P(\neg A \neg B)$
Lift/Interest	$P(B\|A)/P(B)$ or $P(AB)/P(A)P(B)$
Leverage	$P(B\|A) - P(A)P(B)$
Added Value	$P(B\|A) - P(B)$
Relative Risk	$P(B\|A)/P(B\|\neg A)$
Jaccard	$P(AB)/(P(A) + P(B) - P(AB))$
Certainty Factor	$(P(B\|A) - P(B))/(1 - P(B))$
Odds Ratio	$\frac{P(AB)P(\neg A \neg B)}{P(A \neg B)P(\neg BA)}$
Yule's Q	$\frac{P(AB)P(\neg A \neg B) - P(A \neg B)P(\neg AB)}{P(AB)P(\neg A \neg B) + P(A \neg B)P(\neg AB)}$
Yule's Y	$\frac{\sqrt{P(AB)P(\neg A \neg B)} - \sqrt{P(A \neg B)P(\neg AB)}}{\sqrt{P(AB)P(\neg A \neg B)} + \sqrt{P(A \neg B)P(\neg AB)}}$
Laplace Correction	$\frac{N(AB)+1}{N(A)+2}$ N= total number of records
Piatetsky-Shapiro	$P(AB) - P(A)P(B)$
Cosine	$\frac{P(AB)}{\sqrt{P(A)P(B)}}$

Table 2.1: List of interestingness measures [17]

Support or coverage are used to represent the generality of a rule. Confidence or a correlation factor such as the added value or lift is used to represent the reliability of the rule. Many objective measures were used for different applications. The selection of an appropriate measure for a given application is an important issue. There are two methods for compare and analyze measures, ranking and clustering. Both can be based on properties of the measures or empirical evaluations on datasets. Measures can be objective interestingness measures, subjective interestingness measures or semantic measures. The most common are objective measures that are based on probability theory, statistics, and information theory.

There are many algorithms to obtain the association rules. As it can be seen in the Figure 2.3, some of the most used are classified according to the strategy to traverse the search space (breadth-first search or depth-first search) and the strategy to determine the support values of the itemsets. The most used algorithm for association rule mining is the Apriori algorithm. This is an influential algorithm for mining frequent itemsets for boolean association rules.

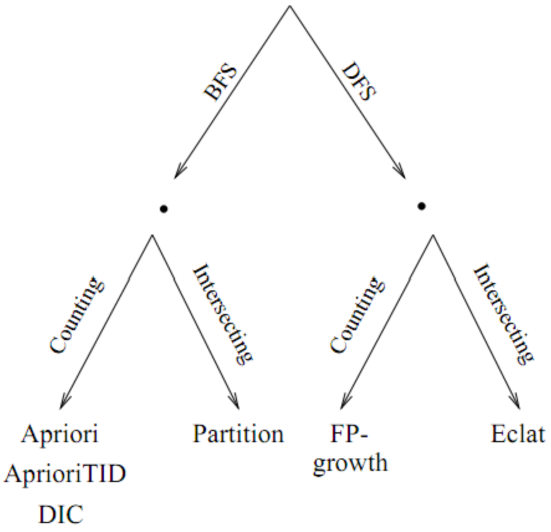

Figure 2.3: Classification of the Association Rule mining algorithms [18]

2.2.4 Uses of Association Rules for Text Mining

Association rules can find correlations between words in one text, for example extract the keywords. In this way some applications of text mining appeared using association rules. In the following sections we explain three examples of this uses.

Extraction Association Rules from Text (EART)

In [25] it is proposed a text mining system called Extracting Association Rules from Text (EART). This system automatically discovers association rules from textual documents. The EART ignores the order in which the words occur, but it focus on the words and their statistical distributions.

The EART system consists of three phases:

1. Text preprocessing phase
2. Association Rule Mining phase
3. Visualization phase

The system accepts a different number of document formats to convert them into XML format. After this, the documents are filtered to discard and ignore the unimportant words. The system also replace special characters with white space between words. After the filtration process the system does word stemming, a process to remove prefixes and suffixes. Then the XML documents are automatically indexed. Each document is described by a set of representative keywords. These keywords help to remember the main theme of the document. The second phase present a way to find information from a collection of indexed documents by extracting association rules from then. EART uses an algorithm for Generating Association Rules based on Weighting scheme (GARW). GARW scans the generated XML document during the generation of the large frequent keyword sets. The last phase extract association rules can be reviewed in textual format or tables, or in graphical format, for example graphs.

Text Mining for Query Refinement

A study in the University of Granada also applies association rules for text mining in the article [13]. This research tries to use the text mining for query refinement. This application helps the user suggesting related words when he is constructing a query. This suggestion could help the user to refine the search and reduce the number of results obtained.

Given a collection of documents, the system tries to find association rules from the terms in the collection. When the user is typing a query and exists an association rule that contains that word in the antecedent or in the consequent, the other term of the rule is suggested to be included in the query. If the term included is the consequent a generalization of the query will occur, while if the term is the antecedent the query will be more specific.

TTM Testbench

Kjetil Nørvåg, Trond Øivind Eriksen and Kjell-Inge Skogstad describe in [30] a process of mining association rules in temporal document databases and they develop an application (TTM Testbench) to work and experiment with this kind of database. A temporal document database is a database with documents that in addition stores old versions of the documents.

The process have some different phases.

1. Tokenization: The first step extracts the terms from the documents. After this step the documents will be transformed into a list of terms.
2. Text filtering and refinement: To obtain rules with more quality some terms that do not contribute are filtered. Remove terms in a stop word list, stemming the terms or analyze the semantic to combine words into one (”Barack” and ”Obama” into ”Barack Obama”) are some of the techniques that can be applied.
3. Association Rule mining: This process uses the First Intra Then Inter (FITI) Algorithm. This algorithm have three phases: Mining large intertransactions itemsets, database transformation and mining large intertransaction itemsets.
4. Rule post-processing: At this point, a lot of association rules are found and it is important to select only those that are interesting. The use of some measures like *support* and *confidence* is a good choice

The Temporal Text Mining (TTM) Textbench is an application to experiment in a temporal document database. In this application a document can be loaded and analyzed with the process previously explained. Some or even all the techniques to filter the text can be applied. The final result will be the association rules generated.

2.3 Generation of Graphs from Texts

As it has been already explained, text mining is a technique to analyze documents and study the relationships between the keywords contained in the text and discover for example the main topic of the text. One possible approach to organize the information obtained from the text is that it can be structured and represented as graphs. This data structure allows us to take in consideration the semantics of the text. The relations between terms can be annotated with a numerical value that will represent if the relation is strong or weak. Most of the work on extracting graphs from texts represent the basic relations between keywords. There are many approaches to generate a graph from a text but basically each text mining instance is represented by a vertex in the graph.

In [28] is proposed the transformation of scientific papers into a graph following these steps:

- Sentences of the text are marked with tags.
- The title of the document and the abstract are filtered.
- The sentences previously tagged are now parsed in order to obtain a syntactic tree.
- This syntactic tree is read node by node and some trees are joined.

Other example can be found in [37] where it is proposed an algorithm to discover useful information from a itemsets database. This algorithm, denominated PAPG (Primitive Association Pattern Generation), propose a method to construct an association graph that will help to discover some types of association rules. Each item in the database is annotated with an integer to decide which items have higher *support* than a minimum decided by the user. The association graph will be constructed with all the items that reach this minimum value. All items have associated a vector of bits with the same length than the number of transactions in the database. Each position of the vector will be set to 1 or 0 depending on if the item appears in the corresponding transaction of the database or not. A directed edge will be created between two nodes i and j (with $i < j$) if, comparing position by position both vectors of bits associated to these nodes ($BV_i \wedge BV_j$), the total number of 1s in the same position of each vector reach the minimum support.

2.4 Topic Modeling

When we have to deal with a large collection of unorganized documents, for example to find the papers related with some topic in a collection of millions of articles, it is not possible with a simple search on the documents, we need a more structured method that helps us.

In this section, we describe topic models, probabilistic models for uncovering the underlying semantic structure of a document collection based on a hierarchical Bayesian analysis of the original text as is described in [6]. The topic modeling can be used in many different collection of documents, a common approach is apply the topic modeling on the abstract section of the scientific documents or simply in a collection of a newspaper articles.

Topic models are a technique that tries to find some useful information in unorganized collection of documents. It discover patterns between words, relate documents that do not seem similar, etc. The basic methodology proposed to analyze a collection of documents consists on compute each document in the corpus and reduce it to a more suitable format. For example, the tf-idf algorithm will obtain a list with the number of occurrences of each word. This frequency will be compared with the frequency in the entire collection. After tf-idf the documents will be reduced to a list of keywords and numbers. Anyway this approach does not represent a significative reduction in the amount of information. Due to this problem, appeared a topic model called probabilistic latent semantic indexing (PLSI) [19].

In 2002, David Blei developed a generalization of PLSI called Latent Dirichlet Allocation (LDA) [6] that is probably the most common topic modeling algorithm. The main feature of LDA is the possibility of extract multiple topics from one document. It is obvious that in a collection of documents there will be many topics and LDA assumes that it will obtain some of these topics with different proportion from each document. This assumption has sense because usually the documents in a corpus tend to be heterogeneous, for example a collection with the documents that belong to a particular conference. Some of the applications of LDA are, for example, explore the collection looking for a particular topic and obtain the documents related with that topic; visualize a document plotting the different topics proportions obtained on it, to see which are the most relevant topics in the document; find similar documents using the topic proportions to define a similarity measure between documents.

The HITS (Hyperlink- Induced Topic Search) algorithm was originally created to study the links between web pages in order to extract and group pages that could be related with some topic. The algorithm discover communities called "authorities" and "hubs". Authorities are pages that are good to obtain information from the information that contain themselves and consequently they will be linked from many other pages. On the other hand, hubs are pages that are good by following the links that contain to other pages. These values are calculated with the following two operations [29]:

$$x_p = \sum_{q,q \to p} y_q$$

$$y_p = \sum_{q,p \to q} x_q$$

Where for each page p, the weight of x_p is the sum of y_q over all the pages that contain a link from q to p $(q \to p)$. Respectively, y_p is calculated as the sum of all x_q.

Chapter 3

Association Networks Construction

This project is divided in two big sections. In this chapter we will describe the first part that introduces our approach to construct an association network. The first section briefly explain the different phases followed in this part of the project. Having a collection of documents, we explain all the steps and methods carried out to get the final association network. Then we will describe some implementations details, nevertheless, in this section we do not show any code because we believe that it is not really important to understand the final results. Finally we will discuss the properties of the association network constructed and conclude if it has small-world networks properties or if it is simply a random graph.

3.1 Approach

This section, in order to give and overview, explains the process followed in this project to construct an association network. It is divided according to the different steps carried out to build the network.

3.1.1 Preprocessing

Text documents can be found in different formats (e.g. pdf, doc or odt). These formats make easy the reading but are not suitable to work with the files due to the amount of unimportant information that they contain. For example different encoding or formatting, font type, size and distribution of the text. To avoid this and focus only in the text, documents have to be transformed into plaintext files. Nevertheless, if documents contains special characters, symbols or formulas; the transformation to plaintext is not always perfect and some undesired words could appear.

3.1.2 Concept Extraction

Once the documents are in a correct format, the next step is the extraction of the keywords of each document. In our work we performed the extraction of the terms with four different existing methods developed in previous researches.

- Manual Annotation: The terms are extracted by humans. The motivation to include this method is to see if the properties of networks based on automatic annotations are similar to those produced by humans.
- Maui Indexer: This method extracts topics using the keyphrase algorithm Kea and uses some other functionalities.
- Wikipedia Miner: That uses Wikipedia collection of articles as vocabulary to extract the terms.
- Topia Term Extract: Uses a part-of-speech tagging algorithm to mark the words with a tag. It will extract the terms with more occurrences.

A database with all the keywords extracted by each one of the methods explained will be created.

3.1.3 Association Rule Mining

Once all the keywords have been stored, the next step consists on extract all the association rules according with an interestingness measure selected. For this

Measure	Formula
Confidende	$P(B\|A)$
Conviction	$\frac{P(A)P(\neg B)}{P(A\neg B)}$
Lift	$P(B\|A)/P(B)$ or $P(AB)/P(A)P(B)$
Conf*Lift	$P(B\|A) * P(AB)/P(A)P(B)$
Leverage	$P(B\|A) - P(A)P(B)$
Jaccard	$P(AB)/(P(A) + P(B) - P(AB))$
Laplace Correction	$\frac{N(AB)+1}{N(A)+2}$ N= total number of records
Accuracy	$P(AB) + P(\neg A\neg B)$
Certainty Factor	$(P(B\|A) - P(B))/(1 - P(B))$
Piatetsky-Shapiro	$P(AB) - P(A)P(B)$
Cosine	$\frac{P(AB)}{\sqrt{P(A)P(B)}}$

Table 3.1: List of interestingness measures implemented in the project [17]

approach we extracted the association rules using different interestingness measures that are detailed in the table 3.1. The reason to do this is because we want to study the effects of different measures on association networks. Depending on the interestingness measure, the association rules extracted will be different and we need to find out which measures are good to obtain an association network with the properties of the small-world networks and not just a random graph.

These interestingness measures can then be used to classify the rules by importance or by any other criterion. It is not possible to see a priori which measure will get the rules that are interesting for our network.

Usually, in association rule mining to filter the interesting rules, a threshold is established and all the rules with a value higher than this minimum value are selected. The problem of use only threshold approach in our work is that with high threshold, many items would be disconnected from the network; if we lower the threshold, then all the items are included at least in one rule, that implies that the number of rules obtained is getting too high. Due to this inconvenient, for our approach, as well as use the threshold, for each item we decided to sort all the rules, having first the rules with higher value for the interestingness measure, and use only the top k rules of this sorted list. Although standard association rule mining considers rules with multiple items (e.g. $(A, B) \rightarrow C$), in our project we only use association rules with only one simple item in the antecedent and the same for the consequent ($A \rightarrow B$).

3.1.4 Graph Construction

Having an item set $I = \{i_1, i_2, ..., i_n\}$ that contains all the concepts extracted from a collection of documents, a set of association rules $R = \{r_1, r_2, ..., r_m\}$ can be obtained. Where r_i with $i \in [1..m]$ is an association rule of the form:

$$r = i_i \rightarrow i_j; \qquad i_i, i_j \in I$$

An association network can be constructed as a graph G using the association rules $r \in R$:

$$G = (V, E); \qquad V \equiv I; E := r \in R$$

In less formal words, the set of vertices V of the graph will represent the words and the set of edges E are defined by the association rules $r \in R$. Each vertex $v \in V$ will have associated two sets of edges, one with outgoing edges $E_o = \{e_{o1}, e_{o2}, ..., e_{on}\}$ and other list with ingoing edges $E_i = \{e_{i1}, e_{i2}, ..., e_{in}\}$. Two nodes ($v_i$, v_j) will be connected by an edge e depending on if they are related or not by any association rule.

$$\exists e \in E \ / \ e = a \rightarrow b \implies \exists r \in R \ / \ r = a \rightarrow b$$

The graph G will have the following properties:

- G is a finite graph.
- G is a directed graph.
- G is a weighted graph.
- G is not a complete graph, not all pair of nodes are connected by an edge.
- Any edge e in G corresponds to a rule r in R with a one-word consequent.
- G is not a regular graph because the nodes can have a different number of incoming edges; the number of outgoing edges k is set by the user.

3.2 Implementation

This section describes with more technical details the process to construct the association network. Although there is a reference to the source code, the results are the most important part of the project and not the implementation.

3.2.1 Simple Graph Database

Simple Graph Database (SGDB)[1] [11] is a Java library to store and process the graph data to construct the association networks. The code we have developed during the work on this thesis has contributed and it has been added to the SGDB project. The Goal of SGDB is to provide decent performance for graph traversal processing even for larger graphs. It is optimized for spreading activation algorithm over small-world networks. Usually researchers and developers working with graphs, they load and process the whole graph in memory. The reason for this is because when the graph is small enough the performance is higher. The problem becomes when the graph is so big that it can not fit in memory.

Association Rule Mining Phase

First of all, the inputs of our system are the concepts extracted from the collection of documents; a set of concepts belonging to a document is considered as a transaction for association rule mining. We will use the association rule mining procedure that is included in the SGDB library. More detailed, we will compute with our procedure all the keywords with an interestingness measure, a file with all the association rules will be created. Each row of the file has the same format containing information of the association rule like the antecedent node, consequent node or weight obtained with the chosen interestingness measures plus others.

Graph Construction

From each association rule contained in the association rules file, an edge connecting the nodes will be created in the graph. The weight of the edge will

[1] http://sourceforge.net/projects/simplegdb

be set to the corresponding in the association rules file. Once the network is built it is necessary to analyze it and obtain some statistics of interest in order to determine if the graph is valid or not. Specifically the properties obtained in this implementation are:

- Number of nodes.
- Number of edges.
- Average clustering coefficient.
- Average path length.

Every node of the graph will belong to a community or cluster.

Degree Distribution

The degree distribution of the network is also very important. In order to study the distribution followed by the graph, the CDF (Cumulative Distribution Function) or just distribution function is saved in other file. A. Cluset, C.R. Shalizi and M.E.J. Newman published an article [12] on power-law distributions in empirical data and implemented the methods that they describe in that article. We used a Matlab implementation of a function[2] developed in the previous research that implements both the discrete and continuous maximum likelihood estimators for fitting the power-law distribution to data. This function extracts the α parameter. This parameter will allow to determine if the graph is a small-world network or simply a random graph The analysis of the properties will be explained with more details in the following section

3.3 Properties of Association Networks

Small-World Networks are graphs that present some special behavior in properties like the average path length, the clustering coefficient or the degree distribution. In order to determine if a specific graph is in effect a small-world network, a good choice is to compare these properties with the ones corresponding to a random graph with the same characteristics.

[2] http://tuvalu.santafe.edu/ aaronc/powerlaws/plfit.m

3.3.1 Properties of Random Graphs

A random graph is a graph that is generated by any random process. One of the models applied in the generation of random networks is the Erdös-Rényi model [15]. In this model any node is connected or not with other nodes with the same probability in all the network. This means that has a statistical independence with the rest of nodes in the network.

3.3.1.1 Clustering Coefficient

The local clustering coefficient for a graph is given by:

$$C_i = \frac{2m_i}{n_i(n_i-1)}$$

Where m_i is the number of edges and n_i is the number of nodes of the subgraph. Since edges are independent and have the same probability p. Then:

$$m_i = p\frac{n_i(n_i-1)}{2} \Longrightarrow p = \frac{2m_i}{n_i(n_i-1)}$$

$$C_{rand} \cong p = \sum_{i=0}^{N} \frac{2m_i}{n_i(n_i-1)}$$

If we represent the average degree of a node as $\langle k \rangle = \frac{2E}{N}$ where E is the expected number of edges in the graph. Finally:

$$C_{rand} \cong p = \frac{\langle k \rangle}{N}$$

3.3.1.2 Average Path Length

Random graphs tend to have a tree-like topology with almost constant node degrees. The number of first neighbors is $N_1 \cong \langle k \rangle$ and the number of second neighbors is $N_2 \cong \langle k \rangle^2$. Accordingly [35]:

$$N = 1 + \sum_{i=1}^{l_{max}} \langle k \rangle^i$$

$$l_{rand} = \frac{\log N}{\log \langle k \rangle}$$

3.3.1.3 Degree Distribution

To calculate the probability $P(k)$ (degree distribution) that a node has k connections in the random network generated with the Erdös-Rényi model. Firstly, it is necessary to calculate the probability p_c that a pair of nodes, chosen randomly, are linked together. To do this, it is calculated the total number of potential pairs in an area of N nodes. This total number is named N_P and its expression is:

$$N_p = \frac{N(N-1)}{2}$$

As the number of pairs linked by the model is M, then the analytical expression of probability p_c can be obtained as:

$$p_c = \frac{N}{N_p} = \frac{2M}{N(N-1)}$$

Taking on the network generated a particular random node v_j, the number of nodes linked in pairs containing to v_j would be $N-1$, because v_j can be linked with exactly $N-1$ remaining nodes of the network. But nevertheless generated links in the M, could be that was not contained v_j. It is assumed then that it was in k of them. The probability in this case that v_j was contained in k pairs of the $N-1$ potential is [16]:

$$P(k) = \binom{n-1}{k} (p_c)^k (1-p_c)^{N-1}$$

This formula follows a binomial distribution for M and N of finite value. Taking into consideration now that the network begins to grow to large values of number of nodes N and links M to reach the point that $N \to \infty$ and $M \to \infty$. In this way we have that the quantity

$$z = \frac{2M}{N}$$

Remains in completely finite values and the degree distribution $P(k)$ becomes a Poisson distribution such as [16]:

$$P(k) = e^{-z} \frac{z^k}{k!}$$

Which as has been mentioned is a Poisson distribution of average z.

However, for most realistic graphs, the power law holds only for a certain range of degrees, namely, for the degrees which not too small and not too large. We will consider the following model with the consideration that most examples of massive graphs satisfying power law have exponent α between 2 and 3. Nevertheless sometimes this value could be higher than 3.

3.3.2 Comparing the Association Network with a Random Graph

In the Figure 3.1 it is represented the evolution of the clustering coefficient and the path length in a graph with p that is the probability that an edge exists or not linking two nodes.

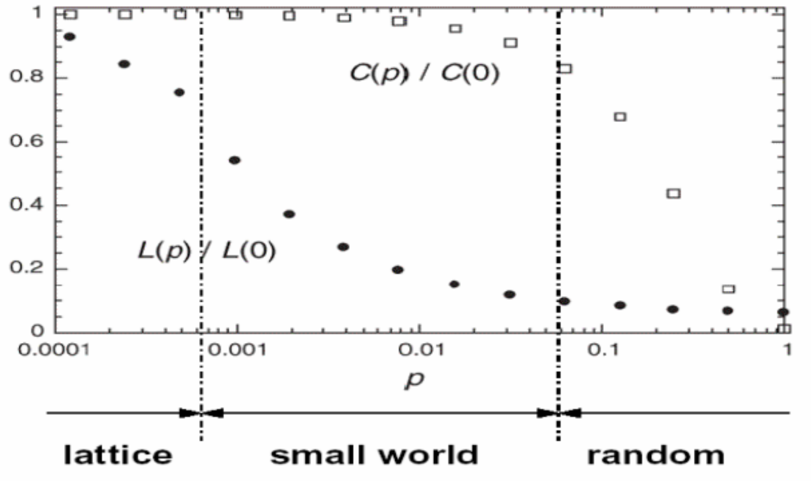

Figure 3.1: Evolution of the average path length (L) and clustering coefficient (C) as a function of the proportion of rewired edges, from regular lattice (leftmost), to small-world networks (center), to random networks (rightmost) [35]

As it can be seen the clustering coefficient and the average path length in small-world networks have behaviors totally different. While the average path length decrease quickly with the probability, the clustering coefficient have a high value until the probability is quite big.

3.3.2.1 Clustering Coefficient

The clustering coefficient in a random graph is low, therefore this property in a small-world network has a higher value.

$$C > C_{rand}$$

3.3.2.2 Average Path Length

The average path length for small-worlds networks has to be lower than in random graphs.

$$l < l_{rand}$$

3.3.2.3 Degree Distribution

If both previously properties achieve the values, the last step to decide if it is finally a small-world network or it just have random properties, is study the behavior of the degree distribution. As it has been mentioned in the previous section, the α parameter has to be between 2 and 3.

3.4 Experimental Settings

This section describes the experiments and the results obtained in our study of the properties of Association Rule Networks. We want to study the impact on the properties of association rule networks depending on the concept extraction method used as well as the impact of different interestingness measures on the same properties

3.4.1 Document Collection

The collection of documents used in this project to construct the graph and study the properties of association networks is the The New York Times Annotated Corpus [33]. We use this collection because it is annotated also manually by humans and we can see whether automatic methods produce similar network structures as annotations by humans.

This collection contains over 1.8 million articles written and published by the New York Times between January 1, 1987 and June 19, 2007 with article metadata provided by the New York Times Newsroom, the New York Times Indexing Service and the online production staff at nytimes.com. As part of the New York Times' indexing procedures, most articles are manually summarized and tagged by a staff of library scientists. Most of the articles contained in the collection are manually tagged by library scientists with tags drawn from a normalized indexing vocabulary of people, organizations, locations and topic descriptors. Some others and even the same articles are algorithmically-tagged. The text in this corpus is formatted in News Industry Text Format (NITF) that is an XML specification that provides a standardized representation for the content and structure of news articles.

3.4.2 Results Analysis

The results analyzed in this section were obtained after build the graph with the collection of the New York Times Annotated Corpus. In this approach we examine all the articles of the year 2005 building one graph per month. In other words, documents are put into groups, one per month of the year and computed to generate the network as previously has been explained.

From each graph, its properties (clustering coefficient and average path length) are extracted and used to generate a chart to compare these values with the same property for a random graph with the same characteristics (number of nodes and edges). This analysis has been done using 4 annotation methods (Manual annotation, Maui Indexer, Wikipedia Miner and Topia Term Extract), 10 interestingness measures (table 3.1) and 12 document collections corresponding each of them to one month of articles published in the New York Times. One month will have approximately 8000 articles. We want to study the behavior on different networks, so the experimentation involves runs over 4*10*12 different configurations.

Manual Anotation

Firstly, Figures 3.2 and 3.3 represent the comparative relation for manual annotations between different interestingness measures. The clustering coefficient is compared in the Figure 3.2 and average path length in Figure 3.3. In the clustering coefficient all the values seem to be dependent on the month and the interestingness measure, but all of them have the same evolution. An important point is the fact that all the values are higher than the corresponding for the random graph. With this annotation method the problem arises looking to the average path length. This property should be lower than in random graphs and the figure shows that all of them are higher.

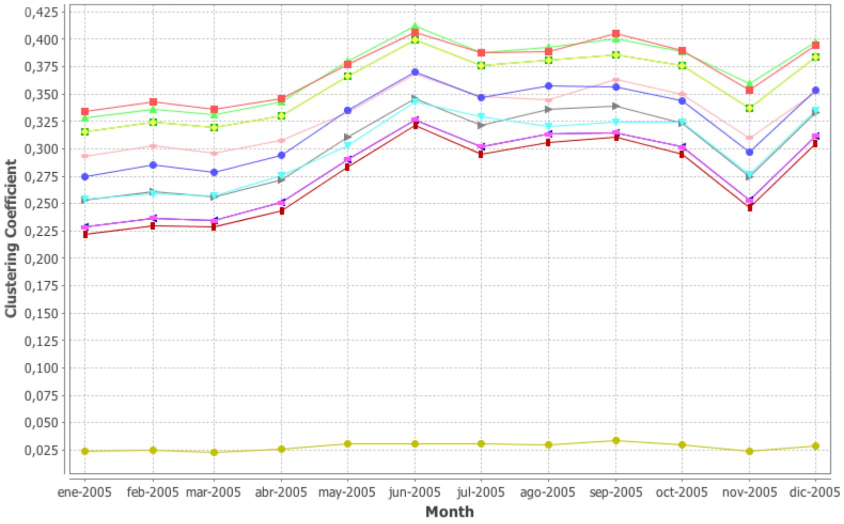

Figure 3.2: Clustering coefficient comparison of interestingness measures for Manual Annotations

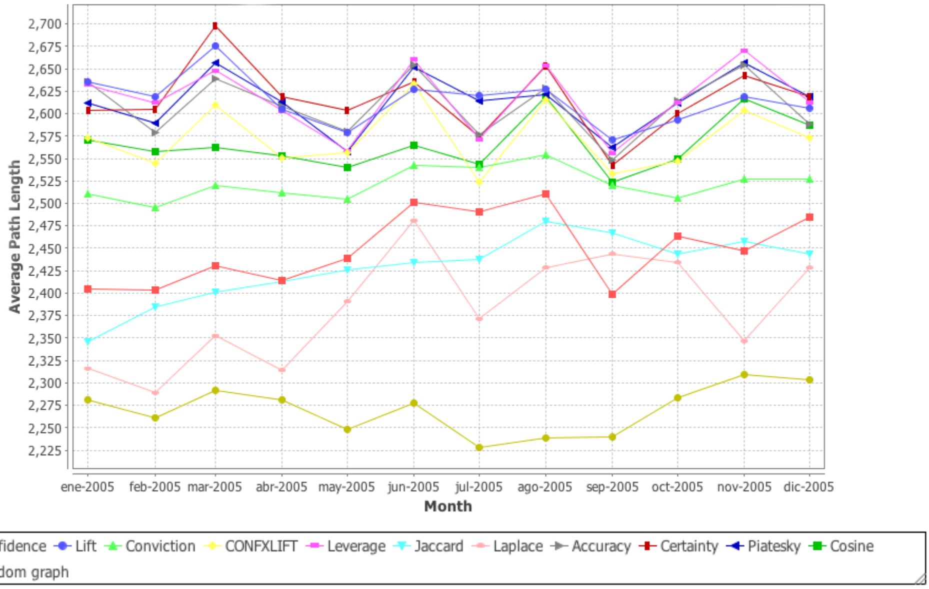

Figure 3.3: Average path length and clustering coefficient comparison between different interestingness measures for Manual Annotations

Maui Indexer

The next annotation method studied is the Maui Indexer. The results of clustering coefficient and average path length are shown in the Figures 3.4 and 3.5 respectively. As in the previous method the results for the clustering coefficient are quite promising for all the interestingness measures compared with the random graph. In this case, in the Figure 3.5 we can extract the first important conclusions about the properties of the graph. As we can see there are three interestingness measures which values are lower than random graphs. These three interestingness measures are exactly *confidence*, *Jaccard* and *Laplace*. This is very promising for our work but in order to determine if the graph is really a small-world network we need to look also at the degree distribution.

In the table of the Figure 3.6 we can appreciate some differences between the interestingness measures. While the values of α are between the theoric desired levels for *confidence* and *Jaccard*, these values never reach the minimum level in the graph built with *Laplace*. According to these values, the interestingness measure of *Laplace* does not achieve the properties to be a Small-World Network with the

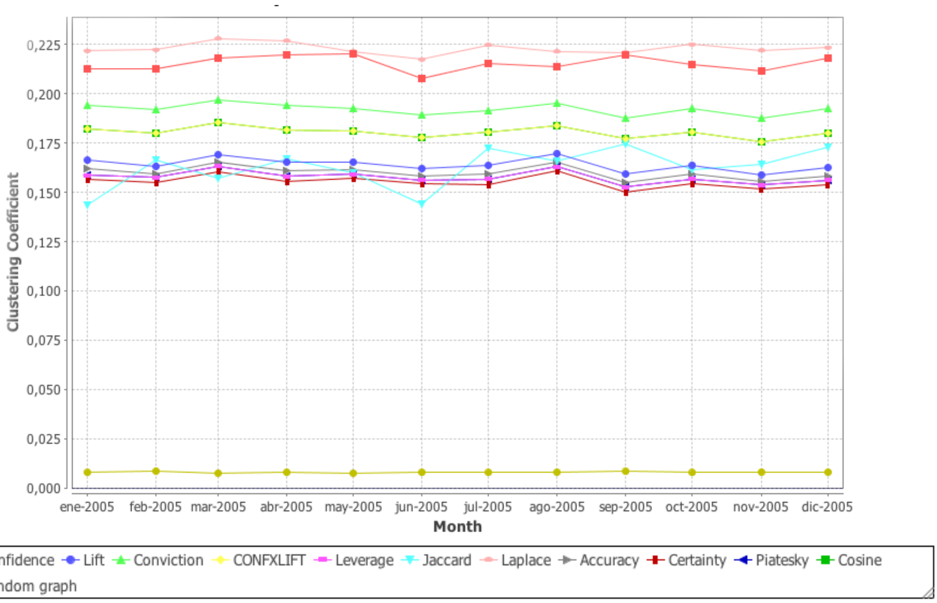

Figure 3.4: Clustering coefficient comparison between different interestingness measures for Maui Indexer

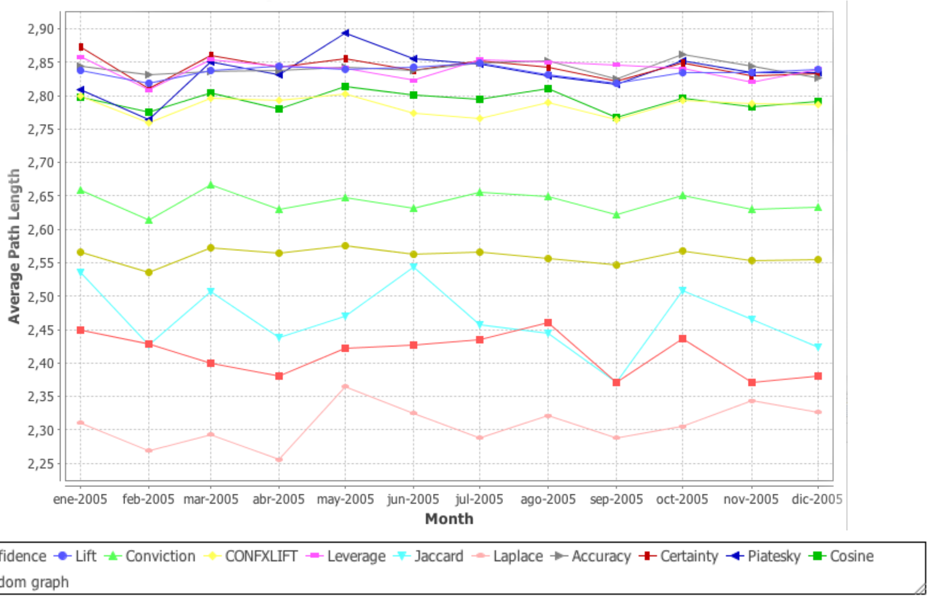

Figure 3.5: Average path length between different interestingness measures for Maui Indexer

Measure	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Confidence	1.97	2.06	2.10	2.00	2.36	1.96	1.95	2.31	1.99	1.95	1.95	2.32
Jaccard	2.51	2.49	2.29	2.30	2.53	2.11	2.56	2.34	2.03	2.42	2.40	2.48
Laplace	1.84	1.84	1.84	1.84	1.83	1.83	1.83	1.86	1.82	1.83	1.85	1.83

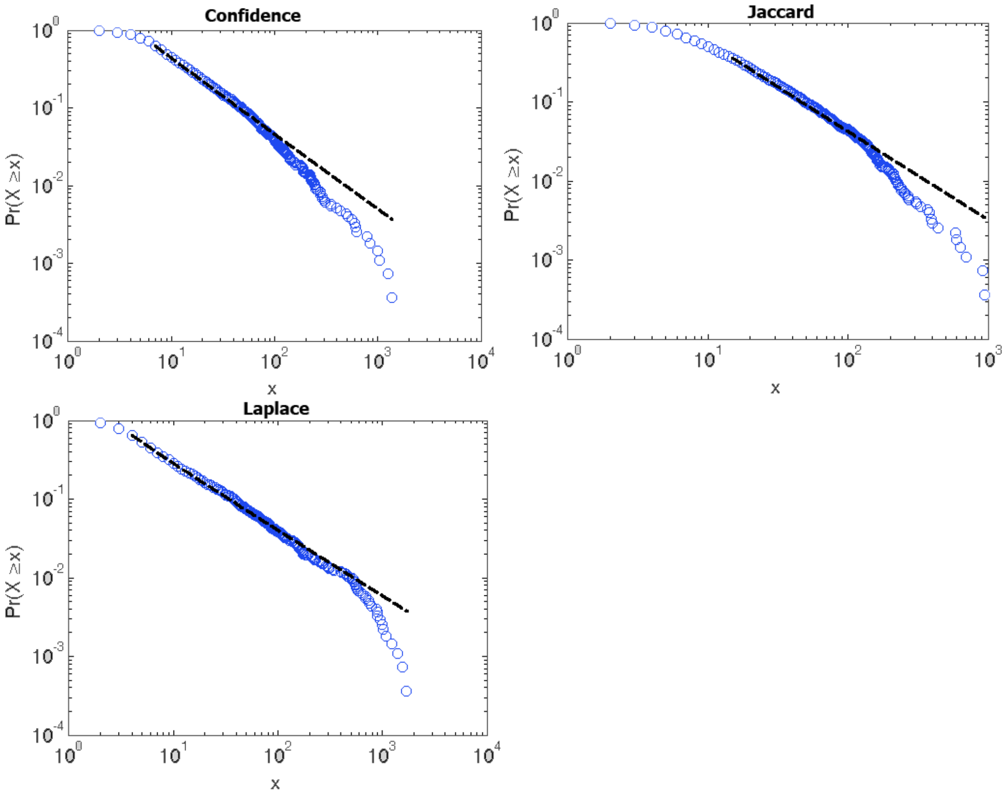

Figure 3.6: Parameter α corresponding to the degree distribution for Maui Indexer

annotations of the Maui Indexer. Anyway, the behavior of the three distributions in the plots (on log-log axes) is not uniform. This is due to the Maui Indexer does not extract many keywords for each file and this implies that the sample is not representative and, even having right values in the table, we do not have enough information to decide if the degree distribution fits with a power law distribution or not.

Wikipedia Miner

The first difference observed after run the Wikipedia Miner to extract the keywords is that with this method, the number of terms obtained for each document is higher than the amount got with Maui Indexer.

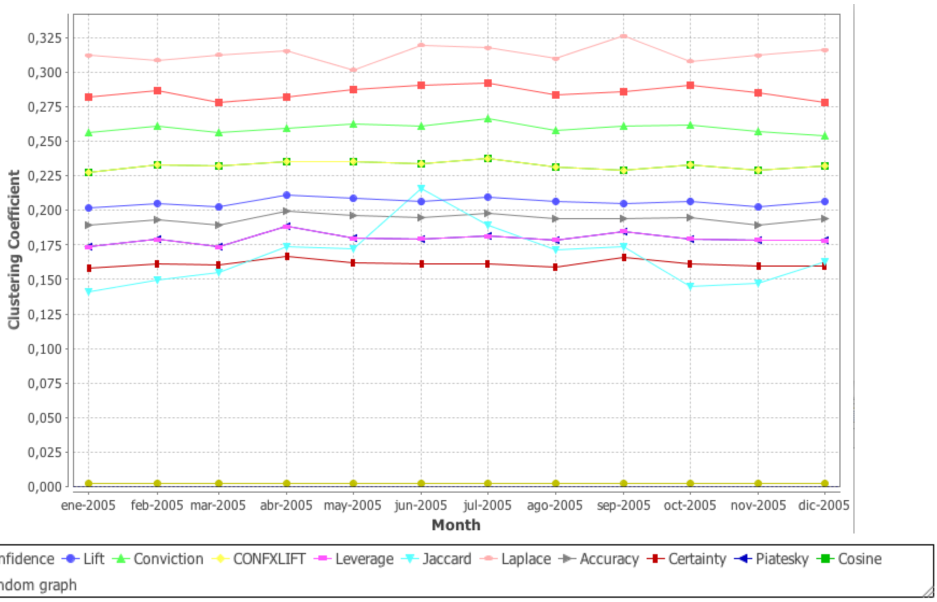

Figure 3.7: Clustering coefficient comparison between different interestingness measures for Wikipedia Miner

The results obtained with the Wikipedia Miner are shown in the Figures 3.7 and 3.8. Looking first to the clustering coefficient, all the values are higher than the corresponding to random graphs regardless of the interestingness measure chosen (see Figure 3.7). Regarding to the average path length, three interestingness measures have values lower than the corresponding to random graphs. These

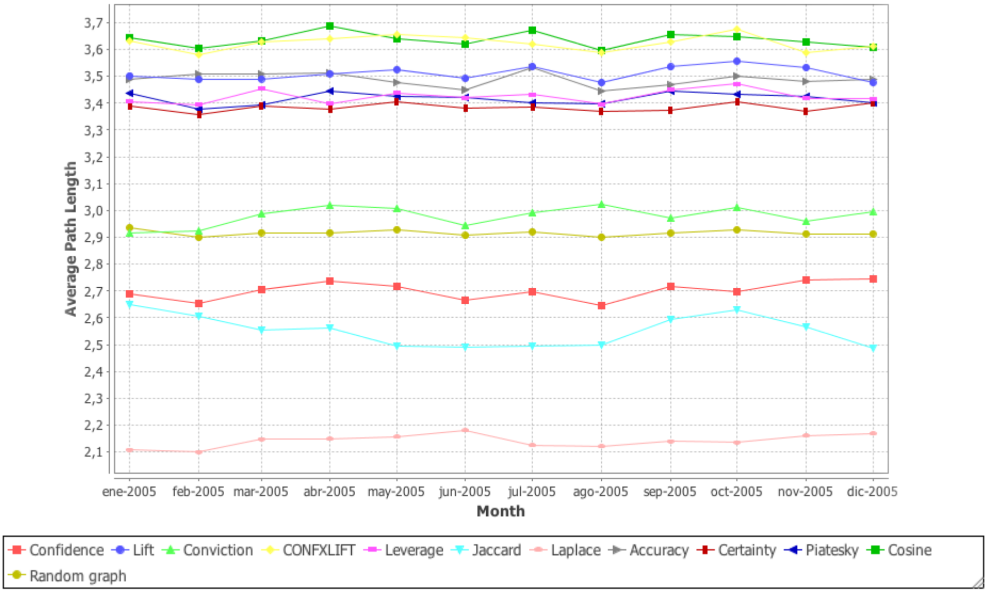

Figure 3.8: Average path length comparison between different interestingness measures for Wikipedia Miner

measures are again *confidence*, *Jaccard* and *Laplace*. This will allow get the first conclusions about the graph. Finally, it is necessary to check if the graphs also match with a power law distribution. These results are displayed in the Figure 3.9 In this case, there are some differences between the Maui Indexer and the Wikipedia Miner when we compare the degree distribution, probably due to that with the number of terms extracted, with this extraction method we can be confident that the results are correct because the sample is big enough to be representative. In the plots of the Figure 3.9 we can observe that *confidence* and *Jaccard* fit with the power law degree distribution, but the behavior of the *confidence* is more constant than in *Jaccard* that some irregularities can be appreciated in the last values of the plot. If we take a look to the table, we can see that all the values of the *confidence* are very uniform and, even without reach the theoretical minimum value, they are always very close to it. In the case of *Jaccard*, some values are in the right range while some others are very low. This can explain the irregularities found in the plot.

Measure	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Confidence	1.79	1.84	1.84	1.79	1.82	2.59	1.89	1.71	1.79	1.77	1.83	1.79
Jaccard	2.17	1.51	2.10	2.22	2.12	2.04	1.92	2.13	2.52	1.51	1.53	2.12
Laplace	1.50	1.50	1.50	1.51	1.51	1.51	1.54	1.51	1.50	1.50	1.51	1.71

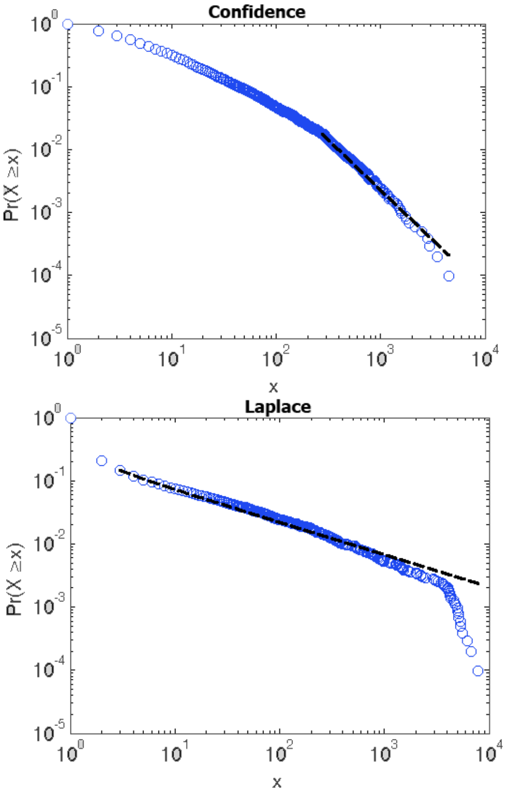

Figure 3.9: Parameter α corresponding to the degree distribution for Wikipedia Miner

Topia Term Extract

The last extraction method is the Topia Term Extract that the first difference observed with the others methods is that the amount of terms extracted with Topia is much bigger than with Wikipedia Miner and with Maui Indexer that, as it has been previously mentioned, extracts few keywords from each document. The

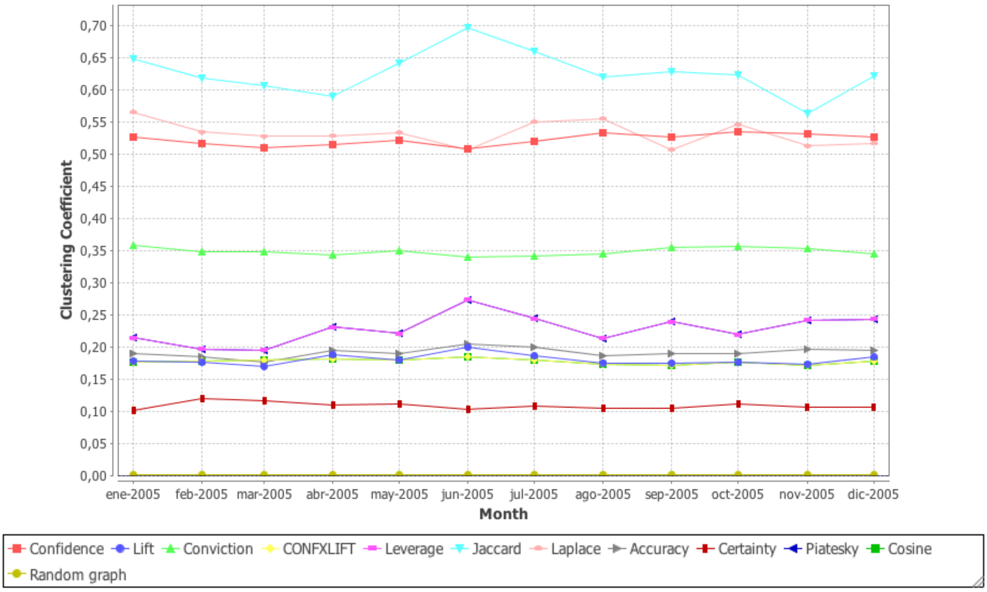

Figure 3.10: Clustering coefficient comparison between different interestingness measures for Topia Term Extract

Figure 3.10 shows the clustering coefficient obtained with Topia and, like in all the other methods, all the interestingness measures get higher values than the random graph. The levels reached with this method are also bigger than the obtained with Maui Indexer and Wikipedia Miner. If we look the clustering coefficient value of the three methods (Y-axis in Figures 3.4, 3.7 and 3.10), for *confidence*, *Jaccard* and *Laplace* Topia is always over 0.50 while Maui and Wikipedia never reach 0.35.

Comparing the average path length we can observe that there are more interestingness measure with a lower value for this property in addition to *confidence*, *Jaccard* and *Laplace*. Since in the other methods these interestingness measures did not achieve these values, we thought that they are not really interesting for our study and we do not take them in consideration to the analysis

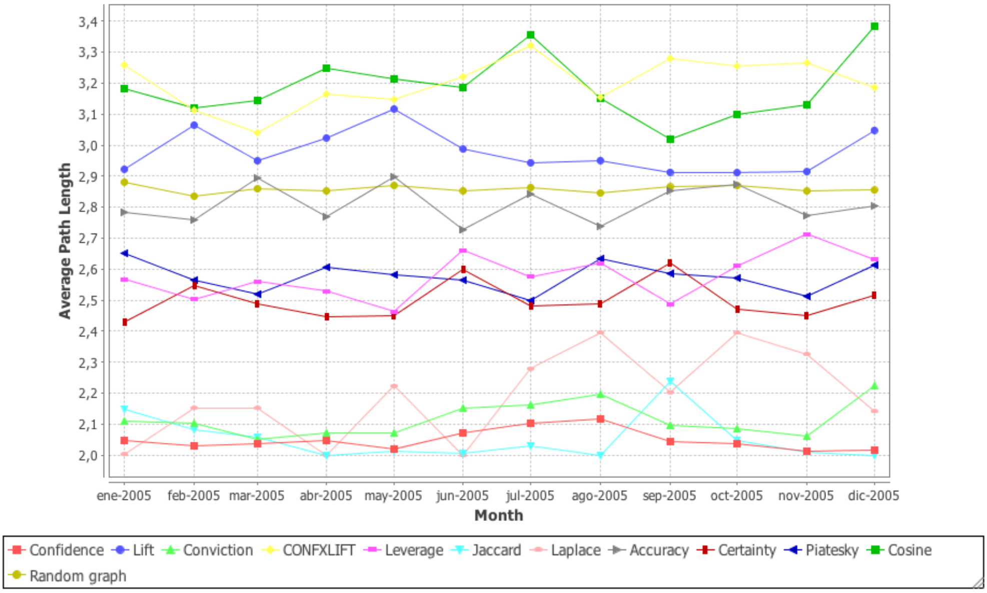

Figure 3.11: Average path length comparison between different interestingness measures for Topia Term Extract

of the degree distribution. We will compare again the behavior of the distribution between *confidence*, *Jaccard* and *Laplace*.

According to the Figure 3.12, we can draw some conclusions about the behavior of the degree distribution. As we can see in the results obtained with Wikipedia Miner (Figure 3.9) the behavior of the degree distribution of the graph constructed with *confidence* was constant while with *Jaccard*, it had some irregularities. These results are confirmed in Topia where, as it can be seen in the plots, *Jaccard* and *Laplace* have very irregular behaviors and only the *confidence* fits with the power law degree distribution.

Measure	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Confidence	1.60	1.60	1.60	1.59	1.65	1.63	1.62	1.68	1.59	1.67	1.61	1.65
Jaccard	1.50	2.45	1.50	1.51	1.50	1.50	1.50	1.50	1.69	1.50	1.50	1.50
Laplace	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50	3.50

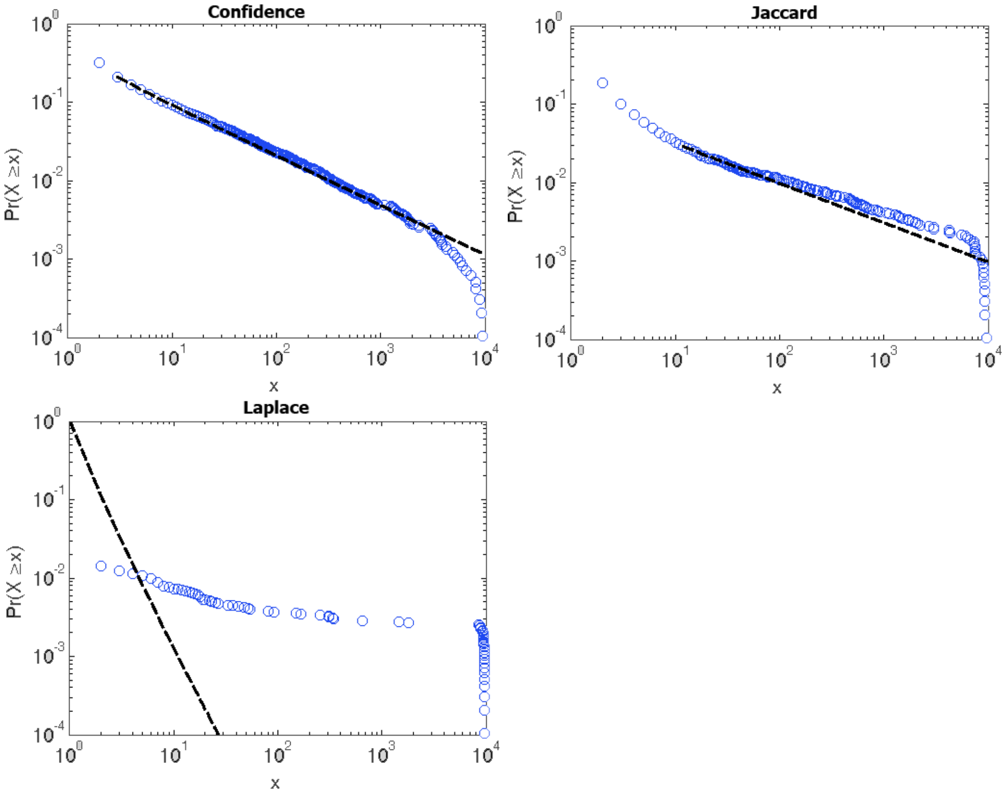

Figure 3.12: Parameter α corresponding to the degree distribution for Topia Term Extract

3.5 Conclusion

The conclusion of the results can be divided in some different parts. We constructed graphs using different annotation methods that have been analyzed in the previous section. We wanted to know if the method used to extract the terms was relevant to construct the association network. As it has been commented before, Maui Indexer extracts only few terms from each document and they are also very general terms. As we can see in the Figure 3.13 the outgoing edges of the node "Spain" are related with very different and general terms. For example this node is directly related with terms like "quixot" and "don quixot" that makes reference to the book written by the spanish writer Miguel de Cervantes. But this node is also connected with many general terms such as "who", "sound" or "book" that are the more important words and they will be also related with other terms that do not have any kind of relation with the original node.

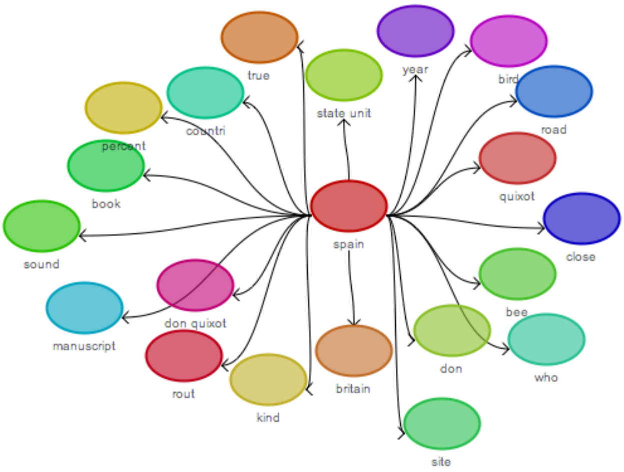

Figure 3.13: Outgoing edges of one node of an association network generated with Maui Indexer

This causes that the association network constructed with the terms extracted by Maui Indexer is not really interesting because the communities obtained does not have sense in most of the times. The association network corresponding to the Wikipedia Miner have better results compared with the previous problems explained for Maui Indexer. In this network, as we can see in the Figure 3.14, using the sense common all the nodes connected with "Spain" are more related with it. Words like "Europe", "United States" or "Government" are the more related words.

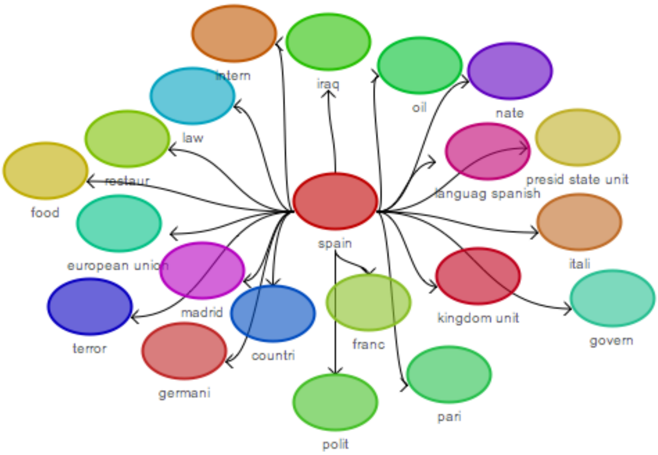

Figure 3.14: Outgoing edges of one node of an association network generated with Wikipedia Miner

The problem of the Wikipedia Miner arises when the domain of the collection is quite specific. We discovered this problem when instead of use the New York Times Annotated Corpus as collection to use the graph, we used a collection of papers from the ACM SIGKDD that is the annual Conference on Knowledge Discovery and Data Mining and contains articles very related with this topic. Wikipedia Miner extracts the terms using the links contained in the Wikipedia articles that are about very diverse topics and some of them do not have relation with the topic of the article. For example when looking for "Data Mining", Topia obtains words always related with the topic while with Wikipedia Miner some terms do not have too much sense like "bear", "classroom" or "university". After study all the different association networks constructed with each method, and taking in consideration that the properties and result of the network will depend on the extraction method used, we decided that Topia Term Extract, obtains the best results in general for our approach, although Wikipedia Miner works quite good when domain of the collection is not very specific.

The second part is to decide which interestingness measure gets the graph with the properties of the small-world networks. As we could see in the previous section only two interestingness measures comply the requirements (clustering coefficient, average path length and degree distribution) when we compare the association network with a random graph. These measures are exactly *confidence* and *Jaccard*. Nevertheless if we take a look to the behavior of each measure in the Figures 3.4, 3.7 and 3.10, we can notice that while all the values of *confidence* are more or less constant for every month, the behavior of *Jaccard* is less uniform and with many oscillations. These irregularities cause that in *Jaccard*, even having right levels of clustering coefficient and average path length, the degree distribution sometimes cannot fit with a power law distribution as it can be seen in the Figure 3.12 where the α values never reach the level and the behavior of the corresponding plot is not the desired.

After this analysis, where we construct an association network using different annotation methods and interestingness measures, we decided that the association network with the best properties is the network constructed using Topia Term Extract to obtain the terms from each document and the *confidence* as interestingness measure to extract the association rules.

Summarizing, different annotation method produce different number of concepts and it has impact on the properties of generated association rule networks. At the same time, different interestingness measures also have a big impact on the properties of the association network.

Chapter 4

Applications of Association Networks

This Master Thesis is focused on the study of the properties of the Association Rule Networks. Nevertheless in this section we discuss some possible applications and in addition, we have tried some initial experiments for topic modeling. Due to the small-world properties discovered on the association rule networks constructed using concepts extracted from a text collection, we can use link analysis (HITS, PageRank) graph mining (community detection) to analyze the text collection.

4.1 Approach

In the previous chapter we explained how we constructed an association network. This network can be used for different applications, between others we cite the followings:

- Representation of a document according to the association network.
- Find relations between communities.
- Topic Modeling.

In the following sections we explain with more details these possible uses of the association network.

4.1.1 Representation of Documents According to the Association Network

The association network constructed contains thousands of nodes with words related to each other. Given a new document, this is analyzed and the keywords are obtained with the same annotation method used to construct the association rule network. Each term of the document of keywords is compared with the association network in order to view the relations of the document with the network. With this information we could find some other documents from the collection related with the new document. A practical application of this approach could be for example, in the webpage of a newspaper each person could have an username and login into a personal page with news. The system, each time that the user reads an article, will analyze it to get the keywords and the next time the user enters in his personal webpage the system will show him the new articles that would be more interesting according to the articles previously read.

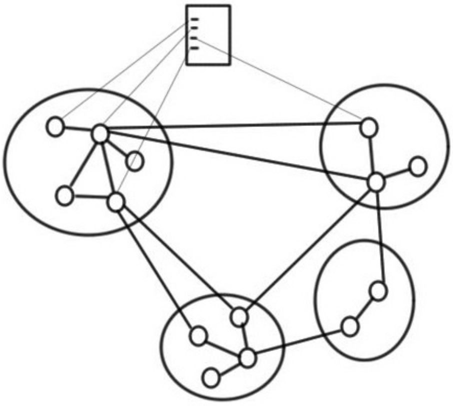

Figure 4.1: Example of association network and relations with a new document

4.1.2 Find Relations Between Communities

All the words contained in the association network are related to each other, but some of then have a stronger link than the others. These words will take part of a community or cluster. In the same way than in the previous example (Figure 4.1) a new document could be compared with the association network in order to discover with which communities is related this document. The interesting point of this is that we can discover that some communities that do not have any kind of relation are, in fact, connected by this document. A practical approach of this is for example, having an association network constructed with the papers of a conference, we can analyze some papers and compare them with the network. The results can show us that there are some researches in different fields of study and in theory not related between them. The previous result will take place if the new document is linked with more than one cluster of the association network. In this way some researches have been held, for example in [31], where performing a *market basket analysis* detecting communities, they discovered relations between items were difficult to find just using association rules.

4.1.3 Topic Modeling

As it has been explained in the Section 2.4, the topic modeling consists on find some useful information in unorganized collection of documents. In this sense, the association network can be used to obtain the main topics of one document and we want to compare the graph based measures against the classical term importance measures (e.g. TF-IDF). The main idea is to check if extracting the main topics of one document using an association network, we obtain better results than with other techniques. The other techniques compared in this study are explained below:

- **Term Frequency (TF):** This is a measure of how often is found a term in a collection of documents or, as in our case, in a single document.
- **Term Frequency - Inverse Document Frequency (TF-IDF):** As it has been mentioned before, this measure will penalize the words that appear in many documents of the collection in addition to the document that is being analyzed.
- **Hypertext Induced Topic Selection (HITS):** It is an algorithm originally created to rank webpages among a collection depending on the links from or to other pages. For our approach this means taking into consideration the ingoing and outgoing edges of every node.
- **Ingoing Edges:** Each keyword of the document is searched in the association network and if exists a node with that term, this measure count the number of ingoing edges of the node.

- **Communities based method:** This is the method implemented using the association network. This method is based on communities, in fact we sum the TF-IDF scores of terms belonging to the same community, after that we take the top 5 communities and output the top 2 terms of each of those communities. The assumption is that semantically similar terms are in the same communities and the goal is to present the user two representative terms from the top 5 communities for the document.

The method carried out to compare the different techniques of topic modeling is the user evaluation. In our opinion, the only way to decide which extracted words are the most related with some text, is asking some people to vote between the different methods.

4.1.4 Communities Detection

The detection of communities is performed with an algorithm called Size Constrained Community Detection (SizConCD) [10]. This algorithm allows the user set the size of the communities generated. Given a node n with a set of edges E, SizConCD will calculate the total sum of the weights of the edges that connect the node n with a cluster C, this values is denominated *affinity*:

$$aff(n,C) = \sum_{i \in C} w(e_{n,i}) : e_{n,i} \in E$$

The node n will be assigned to the community with more *affinity* and if more than one community have the same value, the one with the higher number of members will be chosen.

4.2 Implementation

In the previous chapter we studied and discussed that the best association network is using the Topia Term Extract as annotation method and the *confidence* as interestingness measure. In order to perform this study the first step was construct the association network. We used a collection of papers sent to the ACM SIGKDD (annual Conference on Knowledge Discovery and Data Mining) from 1999 to 2009. Afterwards ten random articles were selected from the collection of the year 2010 and we extracted the top 10 words of each article using all the techniques explained in the previous section. At this moment we want to point that we decided not to include in the evaluation the terms extracted with the HITS algorithm due to the bad results obtained with it since they were always very general terms and in most of the cases the same obtained with the ingoing edges and it could be confusing for the user.

We implemented a very simple web application where we show the abstract of one paper and four lists (Term Frequency, TF-IDF, Ingoing Edges and the Graph Based measure) with the ten terms obtained. Every list appears in random position not to give any hint to the user of which one is each algorithm. Although we only show the abstract of the paper the terms are extracted taking in consideration the entire paper. The reason to do this is because read ten articles would be very tedious for the user and in most of the papers reading the abstract you could have an overview that is enough to decide if the paper is related or not with the topics.

The lists of terms extracted can be ranked from 1 that means that the terms are not related to 5 if the topics are very associated with the paper.

4.3 Evaluation

The user evaluation were performed by very diverse people. Some of them were more or less familiarized with the domain of the papers to rank, like professors and students of Computer Science, while other users were not specialized in the general topic of the conference. The following paragraph correspond to one papers used in the user evaluation and its corresponding topics obtained with the four explained algorithms:

"Compressing social networks can substantially facilitate mining and advanced analysis of large social networks. Preferably, social networks should be compressed in a way that they still can be queried efficiently without decompression. Arguably, neighbor queries, which search for all neighbors of a query vertex, are the most essential operations on social networks. Can we compress social networks effectively in a neighbor query friendly manner, that is, neighbor queries still can be answered in sublinear time using the compression? In this paper, we develop an effective

social network compression approach achieved by a novel Eulerian data structure using multi-position linearizations of directed graphs. Our method comes with a nontrivial theoretical bound on the compression rate. To the best of our knowledge, our approach is the first that can answer both out-neighbor and in-neighbor queries in sublinear time. An extensive empirical study on more than a dozen benchmark real data sets verifies our design. "

TF: graph, linear, edge, neighbor, network, query, bit, 2 K 2, data, compression eurelian.

TF-IDF: linear, 2 K 2, bit, graph, neighbor, edge, neighbor queries, eulerian, compression, query, res.

Ingoing Edges: data, algorithm, number, model, result, method, time, information, example, problem, figure.

Communities Based: neighbor queries, web graphs, compression rates, 2 K 2, undirected graph, lexicographic order, res, log log 1, density threshold, networked world, neighborhood size.

This was the paragraph in which the communities based method obtained the best results in the user evaluation. In this example the communities based method works good because all the other methods extract very general terms while our method, due to the modifier that increases the value of composed terms, got more accurate results. This is just a good example to prove that the communities based method works quite good in some examples. However, as we can see in the Figure 4.2, the final result is not as promising as in the previous article. The results say that the best technique is the Term Frequency, then the TF-IDF and after both, the Communities Based method.

In order to explain these results we introduce other document of the user evaluation where the graph based method obtained very poor ranks.

"How do online conversations build? Is there a common model that human communication follows? In this work we explore these questions in detail. We analyze the structure of conversations in three different social datasets, namely, Usenet groups, Yahoo! Groups, and Twitter. We propose a simple mathematical model for the generation of basic conversation structures and then refine this model

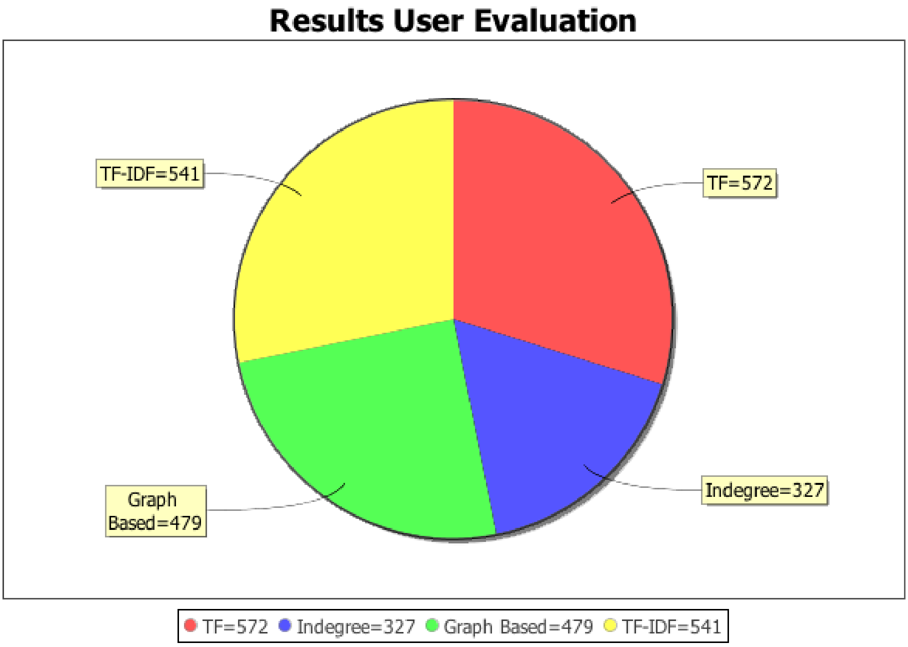

Figure 4.2: Final result of the User Evaluation

to take into account the identities of each member of the conversation. ”

TF: model, thread, group, message, distribution, size, degree, usenet, figure, conversation, process.

TF-IDF: thread, message, group, conversation, usenet, IV deg, IDI, degree, depth, model, distribution.

Ingoing Edges: data, algorithm, number, model, result, method, time, information, example, problem, figure.

Graph Based: degree distribution, modeling process, size distribution, IV deg, blog posts, graph model, 556 proof sketch, graph theory, probability distribution, network information, group data.

The problem discovered of using the association network to perform the topic modeling is that some clusters are too big and not all the words make sense in that cluster. Related with the previous problem, the association network also contains noise in the sense that some nodes are the result of an incorrect behavior of the system that causes that the association network contains words that should not be included. This can be seen in the words extracted by the communities based method in the preceding example. Terms like *"IV deg"* or *"556 proof sketch"* should not be nodes of the network. This could be caused because, at least in this collection, most of the papers contains references to other articles and the extraction method understand these references as if they were normal terms.

4.4 Conclusion

Association rule networks are structures with small-world properties as it has been mentioned before. These properties are very important because that means that they are not just random graphs and they have a special behavior. This confirms, for example, that nodes connected by an edge in the network are more related instead of being randomly connected and that nodes belonging to the same community would be highly connected. Knowing these characteristics we proposed some possible real applications where the association rule network could have a good performance. Due to limitations with time, the only application that we could implement and evaluate was use the association network for topic modeling.

Topic modeling is an important field of research to study the hidden thematic of a collection of documents. We compared classic topic modeling methods with one method based on the communities of the association network constructed. As we could see in the Figure 4.2 the results were not very promising. In the network we could observe that there are some nodes that are just noise. This was caused because the transformation of documents from pdf format to plaintext generate some problematic words due to references, formulas or some other unusual words that can appear on scientific documents. If the documents were already given in plaintext all the nodes with noise would not exist and the relation between the right nodes will be stronger and probably the results would be better because, as we can see again in the Figure 4.2, the graph based method is not that far from the TF and the TF-IDF. Nevertheless a user evaluation is not always reliable, but it is the only method that we found to do this study. Having four lists of words, there is not any automatic method to decide which list is the best. A possible reason that could have caused wrong results was that in our case for example, the texts given to read in the user evaluation were quite technical and difficult to understand. This provokes that if the lists contain similar words, sometimes is confusing for the user and the decision of which list is better is not easy.

Although the use of the association network for topic modeling does not represent a big performance improvement, depending on the application, it can be interesting use it combined with the classical method because the properties of the communities could help to obtain better results. Nevertheless, due to these small-world properties of the network we are sure that association rule networks are very useful in the other application mentioned.

Bibliography

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22:207–216, June 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.

[3] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science (New York, N.Y.)*, 286(5439):509–512, October 1999.

[4] A. L. Barabási, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. July 1999.

[5] A. Barrat and M. Weigt. On the properties of small-world network models. *The European Physical Journal B - Condensed Matter and Complex Systems*, 13(3):547–560, January 2000.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[7] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[8] K. Buss. Literature review on preprocessing for text mining. 2007.

[9] H. Cherfi, A. Napoli, and Y. Toussaint. Towards a Text Mining Methodology Using Association Rules Extraction. *Soft Computing*, 10(5):431–441, 2006.

[10] M. Ciglan and K. Nørvåg. Fast detection of size-constrained communities in large networks. In *WISE*, pages 91–104, 2010.

[11] M. Ciglan and K. Nørvåg. Sgdb: simple graph database optimized for activation spreading computation. In *Proceedings of the 15th international conference on Database systems for advanced applications*, DASFAA'10, pages 45–56, Berlin, Heidelberg, 2010. Springer-Verlag.

[12] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Reviews*, June 2007.

[13] M. Delgado, M. J. Martín-Bautista, D. Sánchez, J. M. Serrano, and M. A. Vila Miranda. Association rule extraction for text mining. In *Proceedings of the 5th International Conference on Flexible Query Answering Systems*, FQAS '02, pages 154–162, London, UK, UK, 2002. Springer-Verlag.

[14] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition, 2005.

[15] P. Erdös and A. Rényi. On the evolution of random graphs. In *Publication Of The Mathematical Institute Of The Hungarian Academy Of Sciences*, pages 17–61, 1960.

[16] A. Fronczak, P. Fronczak, and J. A. Hołyst. Average path length in random networks. *Phys. Rev. E*, 70(5):056110, Nov 2004.

[17] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38, September 2006.

[18] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *SIGKDD Explor. Newsl.*, 2(1):58–64, June 2000.

[19] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[20] J. D. Holt and S. M. Chung. Parallel mining of association rules from text databases. *J. Supercomput.*, 39:273–299, March 2007.

[21] E. Hovy, Z. Kozareva, and E. Riloff. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, EMNLP '09, pages 948–957, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[22] T. Jiang, A. Tan, and K. Wang. Mining generalized associations of semantic relations from textual web content. *IEEE Trans. on Knowl. and Data Eng.*, 19:164–179, February 2007.

[23] W. Jin and R. K. Srihari. Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM symposium on Applied computing*, SAC '07, pages 807–811, New York, NY, USA, 2007. ACM.

[24] O. Madani and J. Yu. Discovery of numerous specific topics via term co-occurrence analysis. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1841–1844, New York, NY, USA, 2010. ACM.

[25] H. Mahgoub, D. Rsner, N. Ismail, and F Torkey. A text mining technique using association rules extraction, 2007.

[26] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 101–110, New York, NY, USA, 2008. ACM.

[27] D. Milne and I. H. Witten. An open-source toolkit for mining Wikipedia. In *Proc. New Zealand Computer Science Research Student Conf., NZCSRSC*, volume 9, 2009.

[28] M. Montes-y Gómez, A. F. Gelbukh, and A. López-López. Text mining at detail level using conceptual graphs. In *Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces*, ICCS '02, pages 122–136, London, UK, UK, 2002. Springer-Verlag.

[29] S. Nomura, S. Oyama, T. Hayamizu, and T. Ishida. Analysis and improvement of hits algorithm for detecting web communities. *Syst. Comput. Japan*, 35:32–42, November 2004.

[30] K. Nørvåg, T. Ø. Eriksen, and K. Skogstad. Mining association rules in temporal document collections. In *ISMIS*, volume 4203 of *Lecture Notes in Computer Science*, pages 745–754. Springer, 2006.

[31] T. Raeder and N. V. Chawla. Market basket analysis with networks. *Social Network Analysis and Mining*, pages 1–17–17, 2010.

[32] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[33] E. Sandhaus. The new york times annotated corpus, 2008.

[34] C. Schmitz, A. Hotho, R. Jäschke, and G. Stumme. Mining Association Rules in Folksonomies. pages 261–270. 2006.

[35] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[36] S. Yen and A. L. P. Chen. An efficient approach to discovering knowledge from large databases. In *Proceedings of the fourth international conference on on Parallel and distributed information systems*, DIS '96, pages 8–18, Washington, DC, USA, 1996. IEEE Computer Society.

[37] S. Yen and Arbee L. P. Chen. A graph-based approach for discovering various types of association rules. *IEEE Trans. on Knowl. and Data Eng.*, 13:839–845, September 2001.

[38] Z. Yuntao, G. Ling, W. Yongcheng, and Y. Zhonghang. An effective concept extraction method for improving text classification performance. *Geo-Spatial Information Science*, 6:66–72, 2003. 10.1007/BF02826953.

[39] X. Zhu. Hidden markov models. *CS769 Advanced Natural Language Processing*, 2010.