NTNU

Innovation and Creativity

Temporal Text Mining

The TTM Testbench

Ole Kristian Fivelstad

Master of Science in Computer Science

Submission date: June 2007

Supervisor: Kjetil Nørvåg, IDI

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

Problem Description

Based on a temporal document database, which contains versions of documents (for example web-pages), we want to use text mining techniques to find hidden information/rules. Examples of these rules can for example be that if one version of vg.no contains the word "bomb", then there is a large probability that the word "terror" will be in a later version.
The assignment consists of studying techniques in temporal data mining, temporal text mining, use/develop these for our domain and implement one or more of these techniques.

Assignment given: 17. January 2007
Supervisor: Kjetil Nørvåg, IDI

Abstract

This master thesis presents the Temporal Text Mining(TTM) Testbench, an application for discovering association rules in temporal document collections. It is a continuation of work done in a project the fall of 2005 and the work done in a project the fall of 2006. These projects have laid the foundation for this thesis. The focus of the work is on identifying and extracting meaningful terms from textual documents to improve the meaningfulness of the mined association rules.

Much work has been done to compile the theoretical foundation of this project. This foundation has been used for assessing different approaches for finding meaningful and descriptive terms.

The old TTM Testbench has been extended to include usage of WordNet for finding collocations, performing word sense disambiguation, and finally extracting higher-level concepts and categories from the individual documents. A method for rating association rules based on the semantic similarity of the terms present in the rules has also been implemented. This was done to try to narrow down the result set, and filter out rules which are not likely to be interesting.

Experiments performed with the improved application shows that the usage of WordNet can help increase the meaningfulness of the rules. One factor which plays a big part in this, is that synonyms of words are added to make the term more understandable. However, the experiments showed that it was difficult to decide if a rule was interesting or not, this made it impossible to draw any conclusions with regards to the suitability of semantic similarity in the rating of the rules.

All work on the TTM Testbench so far has focused on finding association rules in web newspapers. It may however be useful to perform experiments in a more limited domain, for example medicine, where the interestingness of a rule may be more easily decided.

Keywords: Temporal text mining, Association rules, Intertransaction rules, Word sense disambiguation, Document feature extraction, Semantic similarity, Concept extraction

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an introduction to this project. First, the motivation behind the assignment is given. Second, an overview of the project scope and goals is given. Finally, an outline for the rest of the report is presented.

1.1 Motivation

Data mining, and in particular text mining, has attracted much attention in recent years due to the vast amounts of data available, and the rate of growth. Data mining tools can be used to uncover patterns or hidden relations in the available data, and can potentially contribute greatly to business strategy decisions, knowledge bases, and scientific and medical research.

The emergence of data mining tools has come as a result of the natural evolution in the field of information technology. Both computer hardware and database systems technology has seen a steady progress in the past three decades, and as a consequence, a huge number of databases and information repositories have become available. In addition, the emergence and growth of the World Wide Web has made data available regardless of geographic location. These huge amounts of data are too large for humans to comprehend manually, and thus data mining tools for performing automatic data analysis and pattern discovery is of great interest.

In contrast to data mining, where one looks for patterns and knowledge in structured databases, text mining deals with unstructured, or semistructured, textual data such as reports, e-mails or web-pages. This project will focus on a special case of text mining, namely **temporal text mining**. Temporal text mining tries to uncover knowledge and relations in data with a temporal aspect.

This thesis resumes the work performed on the Temporal Text Mining (TTM) Testbench. This is a project started by two other students the fall of 2005, and continued by myself the fall of 2006. This project has laid the foundation for performing text mining on temporal documents, and and an application has been developed for testing various text mining operations.

There are some people who deserve credit for their contributions to this project. First, I would like to thank Kjetil Nørvåg, who is the initiator of this project, for providing continuous support and guidance throughout the project. I would also like to thank Jon Espen Ingvaldsen for taking time to give me valuable insight in various methods and techniques for finding important terms and topics in textual documents.

1.2 Problem Definition and Goals

The assignment text, translated from norwegian, is given below.

> *Based on a temporal document database, which contains versions of documents (for example web-pages), we want to use text mining techniques to find hidden information/rules. Examples of these rules can for example be that if one version of vg.no contains the word "bomb", then there is a large probability that the word "terror" will be in a later version. The assignment consists of studying techniques in temporal data mining, temporal text mining, use/develop these for our domain and implement one or more of these algorithms.*

As can be seen from the assignment text, the main goal of this project is to identify techniques and algorithms for finding hidden information in a temporal database. As mentioned earlier, some work has already been done in this field by two other projects, one which was performed by two other students the fall of 2005 and one which was performed by myself the fall of 2006. It is therefore natural for this project to carry on where the two other projects left off.

The assignment text states that the information sought can be modeled as association rules, on the form "Bomb" ⇒ "Terror". The two earlier projects have focused on *temporal association rules*, which are association rules that incorporate the notion of time. Since much work has been done in the two projects in this field, no other temporal modeling techniques will be discussed in this project.

The result of the two former projects is an application called the TTM Testbench, presented in Section 2.5, this application comes with an algorithm for discovering association rules in a temporal document collection and various operations for performing preprocessing of the text before running the rule mining algorithm.

One of the shortcomings of the current solution, is that the rules found are not very meaningful. The items in the rules mainly consists of single-word terms, and it is difficult for a user to make sense of them. The main focus of this project will therefore be on how to discover semantically rich concepts or topics from the documents, and perform the rule mining on these extracted concepts and topics. Thus a study of relevant techniques and algorithms for doing this will be carried out.

Another shortcoming is that the document collection used in the process is relatively small, it consists of only 37 documents. This number may be too small to find really interesting rules. An important goal will therefore be to gather a larger document collection, and use this new collection in the experimentation phase.

Finally, the potential number of association rules discovered by the rule mining algorithm is huge, this means that the user will have to manually inspect all the rules to find the ones which are really interesting. This project will therefore study techniques and measures for finding the potentially most interesting rules, so that only these rules are presented to the user.

The goals of this project are listed below.

- Gather, and experiment with, a larger temporal document collection.
- Identify relevant techniques and algorithms for extracting meaningful topics and concepts in textual documents.
- Implement one or more of the techniques and algorithms discovered.
- Identify relevant techniques and measures for rating the interestingness of association rules.

- Implement one or more of these measures.
- Identify areas for further research.

1.3 Project Scope

This project will be limited to studying techniques withing text mining, natural language processing and temporal association rule mining. Only topics relevant to this project will be discussed and evaluated. Techniques and algorithms which depend on the manual construction of training data, and a training phase will not be covered by this project due to the relatively short lifespan of the project. However, tools or algorithms which have been pre-trained may be of interest.

The project will make use of the TTM Testbench developed in the previous projects. A study of possibilities for optimization of this application is left as a further study, and only changes needed by the introduction of new operations or algorithms will be implemented.

This project will focus on *semantical* aspects of the preprocessing phase of the knowledge discovery process. That is, techniques and algorithms for identifying and extracting meaningful document features, such as terms or concepts/topics, from the textual documents. The rule mining algorithm already implemented works satisfactory, and will be used in the mining process of this project without any modifications.

1.4 Report Outline

This report is divided into six chapters. A short description of each of the six chapters is presented below for the ease of reading.

Chapter 1 - Introduction
This chapter provides a brief introduction to this project, the motivation behind it, the problem definition and the goals, and the scope of the project.
Chapter 2 - Background
The chapter gives a presentation of subjects of relevance to this project. This includes an overview of both data and text mining.
Chapter 3 - Analysis
The chapter will elaborate on and analyze fields of interest relevant to this project.
Chapter 4 - Implementation
The chapter will present the new Temporal Text Mining (TTM) Testbench.
Chapter 5 - Experiments and Results
The chapter will present the experiments performed in this project, and give an overview of the results.
Chapter 6 - Further Work
The chapter will present some suggestions for further work on the TTM Testbench.
Chapter 7 - Conclusions
The chapter will present a summary of the findings in this project, and a discussion of the lessons learned.

Chapter 2

Background

The purpose of this chapter is to give a presentation of subjects relevant to this master thesis. First, a presentation of data and text mining is given. In addition, an overview of temporal text mining and association rules will be given. Finally, the previous work on the TTM Testbench will be presented to keep this report self-contained.

2.1 Basic Concepts

Before going into details about the background of this project, a presentation of some of the basic concepts which will be used in this chapter, and the rest of the report, will be given. Table 2.1 presents a list of definitions, while Table 2.2 presents the concepts.

Table 2.1: Definitions

Definition	**Description**
$\mathcal{I} = i_1, i_2, ..., i_n$	a set of items
$\mathcal{D}$	task-relevant data
$\mathcal{T}$	a transaction
$\mathcal{A}$	a set of items, A$\subset \mathcal{I}$
$\mathcal{B}$	a set of items, B$\subset \mathcal{I}$

Table 2.2: Basic Concepts

Concept	**Description**
Association Rule	An *association rule* is an implication on the form $\mathcal{A} \Rightarrow \mathcal{B}$, where $\mathcal{A} \subset \mathcal{I}$ and $\mathcal{B} \subset \mathcal{I}$, and $\mathcal{A} \cap \mathcal{B} = \emptyset$.
Support	The *support* of an association rule is the percentage of transactions in $\mathcal{D}$ that contains $\mathcal{A} \cup \mathcal{B}$. This gives, *support*($\mathcal{A} \Rightarrow \mathcal{B}$)=P($\mathcal{A} \cup \mathcal{B}$).

continued on next page

continued from previous page

Concept	**Description**
Confidence	The *confidence* of an association rule is the percentage of transactions in $\mathcal{D}$ containing $\mathcal{A}$ that also contains $\mathcal{B}$. This gives, *confidence*($\mathcal{A} \Rightarrow \mathcal{B}$)=P($\mathcal{B}$|$\mathcal{A}$).
Transaction	A *transaction* $\mathcal{T}$ is a set of items such that $\mathcal{T} \subseteq \mathcal{I}$. Each transaction has a transaction identifier, TID.
Itemset	An *itemset* is a set of items. E.g. $\mathcal{A}$={Beer, Pretzels, Bread} is an itemset.
k-Itemset	A k-itemset is an itemset containing k items. E.g. $\mathcal{A}$={Diapers, Milk} is a 2-itemset.
Minimum support	An itemset satisfies *minimum support* if the occurrence frequency of the itemset is greater then or equal to the product of a minimum support threshold and the total number of transactions in $\mathcal{T}$.
Large itemset	An itemset is large if it satisfies minimum support.
$\mathcal{T}$ contains $\mathcal{A}$	A transaction $\mathcal{T}$ contains an itemset $\mathcal{A}$ if and only if $\mathcal{A} \subseteq \mathcal{T}$.
Text document dataset	In a *text document dataset*, each document is treated as a ”bag” of words.
Term	A *term* is one or more blocks of structured text, for example a word or a phrase.
Tag	A *tag* is a lexical class marker associated with a term.
Stopword	A *stopword* is a common or general term, for example ”this” or ”but”.

2.2 Text Mining

Text mining is a subfield of data mining, this section will therefore first give an overview of data mining before describing text mining. There is some disagreement as to the definition of the term *data mining* [42], [14], [9]. Some people see data mining as one of the (essential) steps in the multi-step process called *knowledge discovery in databases* (KDD). Others use the terms KDD and data mining interchangeably. This report will take the latter approach. That is, data mining and KDD refer to the same process. This process is defined in [10] as *the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*

Data mining as a field has evolved as a result of influence from other disciplines [9]. Figure 2.1 shows some of the most influential disciplines. In addition, data mining has been influenced by the field of multimedia and graphics, because an important step in the data mining process is to be able to present the results to the user in a clear and meaningful manner.

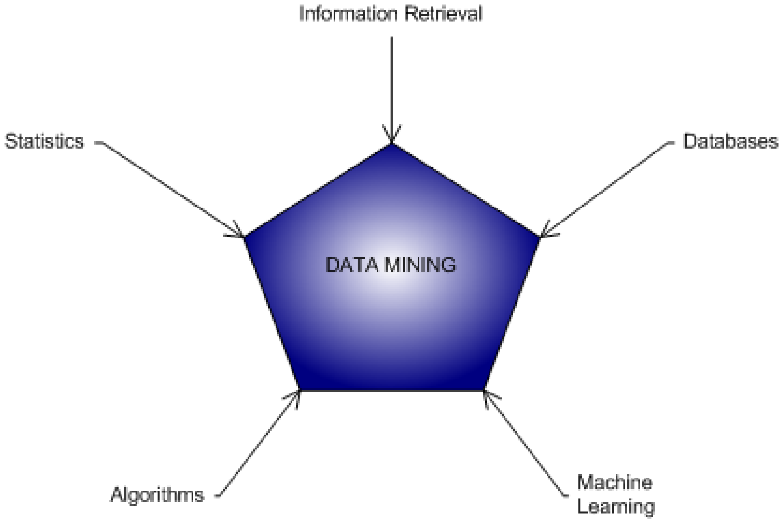


Figure 2.1: Disciplines which has influenced data mining

Data mining has many real world applications, and is reported to have been used in large-scale problems in both science and business [10]. Examples of usage include analysis of customer databases in marketing, portfolio management with regards to future investing, troubleshooting in manufacturing, fraud detection, and even in helping coaches organize and interpret data from basketball matches.

The data mining process consists of a number of essential steps. These steps are listed below.

- **Data Cleaning** - noise and inconsistent data is removed.
- **Data Integration** - data from multiple sources is combined.
- **Data Selection** - the relevant data is retrieved.
- **Data Transformation** - data is transformed or consolidated into appropriate forms.
- **Pattern Mining** - the process where patterns or knowledge is mined.
- **Pattern Evaluation** - the extracted patterns are filtered based on some interestingness measure.
- **Knowledge Presentation** - the patterns found in the previous steps is presented to the user.

The first four steps listed above are steps in which the data is prepared for mining, and the final two steps can be seen as a post-processing step. In the rest of this report, the data mining process will be defined by three steps listed below.

- **Data Preprocessing** - the input data is prepared for further analysis, steps 1-4 above.
- **Pattern Mining** - the step where the knowledge is mined, step 5 above.

- **Data Postprocessing** - the patterns are ranked and presented to the user, steps 6-7 above.

The rest of this section will focus on text mining, also called knowledge discovery in text (KDT). For a more detailed presentation of the field of data mining, I refer to a text-book on data mining, such as [42], [14] or [9].

Work on data mining has usually been focused on mining knowledge from structured databases [12]. The recent years, much research has been made on made on mining knowledge from textual sources due to the vast amount of textual data available, and the need for turning this data into useful knowledge. By extending the definition for data mining given above, the definition of text mining becomes [20]:

> *The non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in unstructured textual data.*

As can be seen from the definitions of data and text mining above, these are closely related fields, the difference lies in what kind of data is used in the knowledge discovery process. The steps performed in the process are therefore quite similar. Both systems rely on some kind of preprocessing, pattern-discovery algorithms and a presentation layer [11]. However, since data mining usually assumes that the data is stored in a structured format, the focus of preprocessing lies mostly on scrubbing and normalizing the data. In text mining on the other hand, preprocessing centers around the task of identifying and extracting representative features of the textual documents. This preprocessing is responsible for transforming unstructured text into a more explicitly structured format.

Because natural language text plays such a central part in text mining, this process draws on advances in fields concerned with the handling of natural language. In particular, text mining makes use of techniques and algorithms in the areas of information retrieval, information extraction, and corpus-based computational linguistics [11].

The main tasks in a text mining system are similar to those of a data mining system, presented earlier. An overview of the high-level functional architecture of a text mining system is shown in Figure 2.2. As can be seen from the figure, the system takes as input a raw document collection, this collection is then processed and transformed into a form appropriate for further analysis. This involves, among other things, identifying and extracting the relevant features to represent the documents. The next phase is the execution of the core mining operations, and presentation of the results to the user. The main task in this phase is the actual mining for knowledge. The user will usually be involved in this phase, both by specifying which patterns to look for and which parameters to use in the mining process. In addition, the user must be able to view and browse the results.

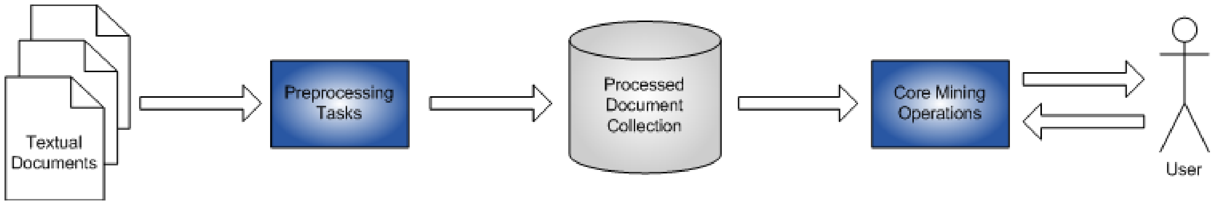


Figure 2.2: High-level functional architecture of a text mining system, adapted from [11]

2.2.1 Document Collections and Documents

The key element in text mining is the *document collection*. These document collections may be either static (the initial set of documents remains unchanged) or dynamic (new or updated documents are added regularly). The main element of the document collections are the documents themselves. A document can informally be defined as *a unit of discrete textual data within a document collection that usually, but not necessarily, correlates with a real-world document* [11]. Examples of such documents may be e-mails, reports, research papers, news stories, articles or even patient journals.

The documents in a document collection are usually ”weakly structured” or ”semistructured”, even though they are often, somewhat misleading, labeled as completely unstructured. Even a rather short and insignificant document may demonstrate much structure from a semantic and syntactical view. Also, elements such as punctuation marks, capitalization and special characters can often serve as ”soft markup”. Therefore documents with nothing less then typographical elements as structure, are referred to as ”weakly structured”, examples of such documents are business reports or news stories. At the other side of the spectrum are documents where some meta-data may be inferred due to the use of extensive and consistent format elements. These documents can be described as ”semistructured”. E-mails, HTML web pages and PDF files are examples of ”semistructured” documents.

2.2.2 Document Features

The preprocessing phase in text mining seeks to transform an irregular and implicitly structured representation into an explicitly structured representation. Documents can have a potentially huge number of words, phrases and sentences, and these elements may have a large number of senses in various contexts and and combinations. One of the most essential tasks for the system is therefore to identify a subset of the document features to represent the document as a whole [11]. This set of features is called the *representational model* of a document.

There are four commonly used document features [11]. Which feature to use depends on the application of the text mining system, the features vary with regards to their amount of semantic expressiveness, their efficiency in the computation, and how easily they are generated. The different features are listed and described below, Figure 2.3 shows an example of each of the four features.

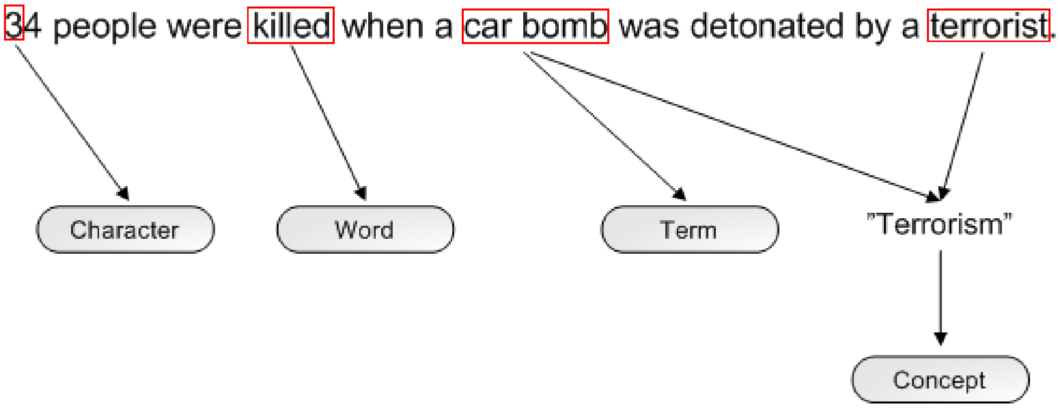

34 people were killed when a car bomb was detonated by a terrorist.
Character
Word
Term
"Terrorism"
Concept

Figure 2.3: Examples of the different types of document features

- **Characters** - These are the individual building blocks in the documents, such as letters, numeral or white spaces. Character-based approaches are usually of very limited value in text mining applications, especially without any positional information. With regards to text processing techniques this approach can be quite troublesome, since very little optimization is done.

- **Words** - Words consists of characters, and are at the basic level of semantic richness. Words are sometimes referred to as existing in the *native feature space* of a document since they are selected directly from the documents. Generally, word-level features should consist of only one linguistic token, thus no multi-word expressions or multi-word hyphenates are considered word-level features. Usually, some optimization is performed, such as removing stopwords[1].

- **Terms** - Terms are single- and multi-word phrases selected from the documents using term-extraction methods. Term-level features are thus made up of words and expressions found in the native document. A term-based representation can therefore be seen as a subset of the words in the document. Term-extraction methods can convert the documents into a series of normalized terms, sequences of tokenized and lemmatized word forms. Various approaches exist for generating and filtering a list of meaningful candidate terms from the set of possible terms for a document.

- **Concepts** - Concepts are generated by means of manual, statistical, rule-based or hybrid categorization methods. These methods proceed by analyzing the word- and/or term-level features in the document and generates one or more suitable concepts. A concept-level feature representation is the only representation where a document may be represented by features not found directly in the native documents. Categorization methods may use an external knowledge source in the process, for example a manually annotated collection of documents. This representation has the features with most semantic value, and is best at handling polysemy and synonymy, presented in Section 3.1. However, this representation has two drawbacks. First, the complexity of extracting and validating concept-level features, and second, the domain-dependency of many concepts.

Note that in the rest of this report, terms and document features will be used interchangeably to refer to the items extracted from documents. That is, terms and document features will both mean any of the three last features shown above, either a single word, a phrase, or a concept. Character-level features are not relevant to this project, and is thus not included.

2.2.3 Core Data and Text Mining Tasks

The main part of a data or text mining system is the task being performed during the analysis phase. The kind of pattern found will depend on which task is used. Sometimes, the user may not know in advance what kind of pattern might be interesting, it may therefore be useful for the mining system to be able to perform more than one task.

The core data and text mining tasks can be classified into two categories [14]: **descriptive** tasks, which characterizes the properties of the current data, and **predictive** tasks, which analyzes the current data to be able to make predictions on new data.

- **Characterization and Summarization** - The goal of this process is to summarize the general characteristics of a target class of data. For example produce a description of

[1] A *stopword* is a common or general word, for example ”that” or ”is”.

customers who spend more than 500 NOK in a store in a given period. This process can therefore be labeled descriptive.

- **Classification** - This process tries to find a model which can describe and distinguish class-labeled objects by analyzing a set of training data. The model can then be used to predict which class an object with an unknown class label belongs in. An application of classification is to develop a classifier for predicting the credit rating of new customers in a bank. This process is therefore predictive.
- **Clustering** - Clustering is similar to classification, but unlike classification, clustering performs the analysis on data objects without a class labels. Clustering is based on two principles, maximizing the similarity between objects in the same class, and minimizing the similarity between objects in different classes. An example of usage of clustering can be to group customers in a store by their location for use in marketing. Clustering is a descriptive process, since it finds relationships among the data already present.
- **Outlier Analysis** - This process tries to discover data objects which do not comply with the general characteristics or behavior of the dataset. Outliers can be found using statistical tests, or distance measures where objects that are far from other clusters are considered outliers. Outlier analysis can be useful when trying to discover fraudulent credit card usage by comparing purchases for large amounts with regular purchases by the same account. Outlier analysis is a descriptive process.
- **Association Rules** - Association rules are concerned with frequent patterns, that is, patterns which appear frequently together in the dataset. An example of such a pattern can be that customers that buys apples, also tends to buy oranges. This can be modeled as an association rule; ”Apples” $\Rightarrow$ ”Oranges”. Association rules are therefore useful for marketing and product placement in stores, this is sometimes referred to as **market basket analysis**[9]. Association rules are descriptive since they model the relationships between data objects already present in the dataset.

Since the previous work in this project has focused on association rules, this project will follow the same path. No further description is therefore given of the other knowledge discovery processes. Association rules will be described in more detail in Section 2.3.

2.2.4 Temporal Documents

A temporal document is a document with a related temporal attribute. This attribute can be a version number, a time stamp, or some other attribute. When looking at more than one document, this attribute will indicate the order of, and distance between, the documents. These documents can then be grouped together, and used in the text mining process to find relationships or patterns.

Seen in the light of this, most documents can be described as temporal since they usually have a time stamp related to them, for example the time it was created. The Internet for example, is a source for large quantities of documents which can be considered temporal. Some web pages are continuously updated with new content, and by logging the web page at regular intervals it is possible to create a temporal document collection. Other temporal documents can for example be patient journals, which can be updated with new information about diseases, treatments, and so on.

Temporal documents may be stored in a temporal document database. Temporal databases differ from regular databases in that the data or documents in the database have a temporal attribute. Well known examples of temporal databases are CVS and RCS, as cited in [32]. These databases are usually used for storing versions of source code. The V2 temporal database however, focuses on the storage, retrieval and querying of temporal documents [33].

2.2.5 Temporal Text Mining

This section will focus on a special field of text mining called temporal text mining. Since this field incorporates aspects from temporal data mining, as well as text mining, a brief introduction to temporal data mining will be given before temporal text mining presented.

Temporal data mining is a special case of data mining, and differs from regular data mining in that it performs analysis on temporal data. Temporal data mining is in [22] defined as *data mining of large sequential data sets*. Sequential data means data which is ordered by some index, for example a time stamp. This means that the data is seen in the context of time, and the data may either belong to a single point in time, or to a time period. For a more detailed presentation of temporal data mining, I refer to [22] and [40].

Temporal text mining can be defined as the *discovery of temporal patterns in text information collected over time* [28]. That is, temporal text mining is similar to temporal data mining. Both processes try to find patterns in data where the aspect of time is incorporated, the difference lies in the kind of data mined. This project will focus on the discovery of **temporal association rules**, which will be presented in Subsection 2.3.4.

2.3 Association Rules

As shown in Subsection 2.2.3, there are several possible tasks to perform in the text mining process. However, the focus of this project is the discovery of association rules. A presentation of association rules and how to find such rules, will therefore be given in this section.

An association rule is an implication on the form $\mathcal{A} \Rightarrow \mathcal{B}$, where the presence of $\mathcal{A}$ is likely to imply the presence of $\mathcal{B}$. Association rules can be classified into two types; **traditional association rules** and **temporal association rules**. The differences between these types will be illustrated later in this section.

An important aspect with association rules is that the uncovered relationships are not inherent in the data, and do not represent causality between the items, association rules only detect common usage of items in the dataset [9]. This means for example that if one is mining for association rules in the medical domain, and finds an association rule which states the following ”Allergy medicine” ⇒ ”Measles”, this does not necessarily mean that allergy medicine causes measles. It can however, be the basis for further research to see if it is actually the case.

The way the data is represented has an effect on the discovered association rules. Mining for association rules usually require that the data be modeled as transactions, and each transaction consists of one or more items. This is illustrated in Figure 2.4. Textual data however, is not transactional by nature. The previous work in this project has found a solution for this. Even though textual data is not considered transactional, it is possible to model it as transactions. The way this is done, is by letting each document represent a transaction and the document features are the items in the transaction. This makes it possible to mine for association rules among the document features.

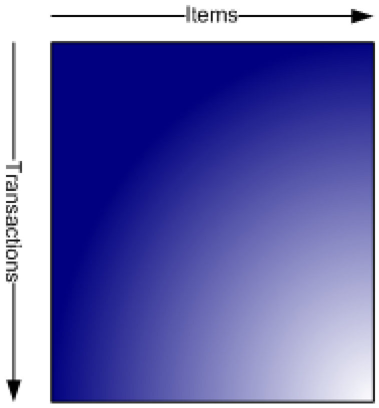


Figure 2.4: A Dataset

2.3.1 Support and Confidence

Association rules are usually ranked by their **support** and **confidence** values. In addition, the support value is used in the rule mining algorithm to prune the possible set of association rules presented to the user. Confidence on the other hand, is not used in the rule mining algorithm itself, but it is used after the initial set of association rules is found to narrow down the result set.

Support

The support of an association rule $\mathcal{A} \Rightarrow \mathcal{B}$ is defined as the percentage of transactions in which both $\mathcal{A}$ and $\mathcal{B}$ appear [9]. That is, the probability of the union of the itemsets $\mathcal{A}$ and $\mathcal{B}$, P($\mathcal{A} \cup \mathcal{B}$).

The support value of an association rule gives an indication of the rarity of the rule. A support close to 1 (or 100%) indicates that the rule is always present, rules which appear this often are most likely already known by the user. A support close to 0 (or 0%) indicates an unusual rule, this can both mean that the rule is interesting, or it could be the result of noise in the dataset.

Confidence

The confidence of an association rule is the ratio of the number of transactions that contains both the antecedent and the consequent of the rule to the number of transactions that contain the antecedent [9]. This is equal to the probability of the consequent of the rule being true, given that the antecedent is true.

$$confidence(\mathcal{A} \Rightarrow \mathcal{B}) = \frac{P(\mathcal{A} \cup \mathcal{B})}{P(\mathcal{A})}$$

The confidence of an association rule gives an indication of the strength of the rule. A confidence of 1 (or 100%) means that if $\mathcal{A}$ is present in a transaction, so is always $\mathcal{B}$. Rules with low confidence on the other hand, may be present only by chance and are probably not interesting.

2.3.2 The Association Rule Mining Process

The process of mining for association rules consists of three main tasks, shown in Figure 2.5. The **text preprocessing** task deals with the detection and extraction of relevant document features from the individual documents. The next task, **rule mining**, is the task where association rule mining algorithms are used to generate association rules. Finally, **rule rating** tries to rate the discovered association rules, so that only the most relevant and interesting ones are presented to the user.

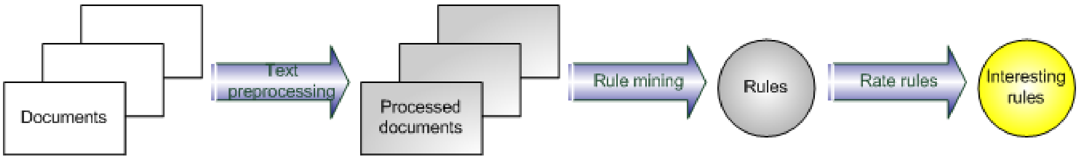


Figure 2.5: The association rule mining process

Operations and methods for performing preprocessing on textual documents by extracting meaningful terms will be described in Chapter 3. As mentioned earlier, this project will use the rule mining algorithm already implemented in the TTM Testbench, an overview of algorithms for mining association rules can be found in Section 2.4. Finally, measures and methods for filtering association rules is given in Section 3.6.

2.3.3 Traditional Association Rules

Traditional association rules are rules on the form $\mathcal{A} \Rightarrow \mathcal{B}$ with support of s% and confidence of c%, where $\mathcal{A}$ and $\mathcal{B}$ are items present in the **same** transaction. An example of such a rule after performing association rule mining on transaction data from a grocery store can be ”Beer” ⇒ ”Pretzels” with support of 30% and confidence of 90%. This can be translated as: *90% of all people who buy beer also buy pretzels.* This kind of association rule can thus be valuable for marketing and product placement.

Table 2.3 presents some example data. Each row in the table corresponds to a transaction. As can be seen in the table, the item ”Milk” is present in 3 of the 5 transactions, this means that the support of ”Milk” is 3/5, which is 0.6 (or 60 %). Similarly, the item ”Ice-cream” is only present in 1 of 5 transactions, which gives a support of 0.2 (or 20%). An example of a rule found from this data can be ”Milk” ⇒ ”Diapers” with a support of 60% and confidence of 100%, which means that both ”Milk” and ”Diapers” are present in 3 of 5 transactions (60%), and every time ”Milk” is present, so is ”Diapers”.

Table 2.3: Sample Transactions

Transaction	**Items**
t_1	Milk, Diapers
t_2	Milk, Diapers
t_3	Ice-cream, Potatoes
t_4	Beer, Pretzels
t_5	Milk, Diapers
...	...

The process of finding association rules can be broken up into two tasks [9], [14], given below. A **large (or frequent) itemset** is a itemset whose number of occurrences (or support count) is above a given minimum support threshold.

1. **Find all large itemsets.**
2. **Generate association rules from the large itemsets found in 1.**

The second task, generating the association rules, is seen as trivial, and a much used algorithm for this task is the Apriori-gen algorithm. The focus is therefore on the first task, finding all large itemsets. A well-known algorithm for this task, is the Apriori algorithm, presented in Subsection 2.4.1.

2.3.4 Temporal Association Rules

Temporal association rules differ from traditional association rules by the fact that they try to model temporal relationships in the data.

There are various types of temporal association rules, these are **intertransaction rules**, **episode rules**, **trend dependencies**, **sequence association rules** and **calendric association rules**. The previous work on the TTM Testbench has focused on intertransaction rules, the other types will therefore not be discussed any further here. For an overview of these, please see [9].

Intertransaction Rules

Traditional association rules only look at items occurring together in the same transactions, and can therefore be called **intratransaction association rules**. There are however, situations where it would be useful to have association rules which span transactions. For example, it would be of interest for a store manager to know that if a customer purchases a computer, they are likely to buy a printer at a later time.

Algorithms for mining intertransaction rules usually utilize a time window. This time window specifies the maximum number of transactions an association rule may span. A time window of 0 means that the algorithm only finds association rules containing itemsets present in the same transaction. Intertransaction rules may therefore be seen as a special case of intratransaction rules [26].

The first algorithms for mining intertransaction rules were E-Apriori and EH-Apriori. These algorithms are based on the Apriori algorithm 2.4.1, and are described in [26]. A further development is the FITI (First Intra Then Inter) algorithm, this algorithm was presented in [44]. The FITI algorithm is developed specifically for mining intertransaction rules, an overview of the algorithm will be given in Subsection 2.4.2.

An example of a rule produced by the FITI algorithm can be {Computer, 0} ⇒ {Printer, 1} with confidence of 80% and support of 10%, which means that 80% of the customers who buy a computer, buys a printer the next day (or another time unit). When comparing this kind of association rule to the problem description given in Section 1.2, one can see that these rules are suitable for the types of rules sought in this project. As an example, consider the rule {Bomb, 0} ⇒ {Terror, 1}. When mining versions of a web newspaper this rule states that if one version contains the word ”bomb”, then the next version is likely to contain the word ”terror”.

2.4 Algorithms for Mining Association Rules

This section will give an overview of algorithms for mining association rules. An algorithm for mining traditional rules, the Apriori algorithm, will be described in Subsection 2.4.1. And the FITI algorithm for mining intertransaction association rules is described in Subsection 2.4.2.

2.4.1 Apriori

Apriori is a well-known algorithm for mining traditional association rules, and is much used in commercial products [9]. The algorithm was first introduced in [1], and is built around the following property: *Any subset of a large itemset must be large.* This means that for the itemset $\{\mathcal{A},\mathcal{B}\}$ to be large, both subsets $\{\mathcal{A}\}$ and $\{\mathcal{B}\}$ must be large.

The Apriori algorithm works in a level-wise fashion. This means that it starts by finding large 1-itemsets (an itemset containing 1 item), these large 1-itemsets are then combined to find 2-itemsets, and so on. This process iterates until there are no more large itemsets to be found. After all frequent itemsets have been found, the Apriori-gen algorithm can be used to generate association rules of the large itemsets.

The process of generating large itemsets is illustrated by Figure 2.6. The bold edges indicate large itemsets. Reading from top to bottom, one can see how the algorithm first discovers large itemsets of size 1, then it proceeds by combining the large 1-itemsets to find large 2-itemsets, and so on.

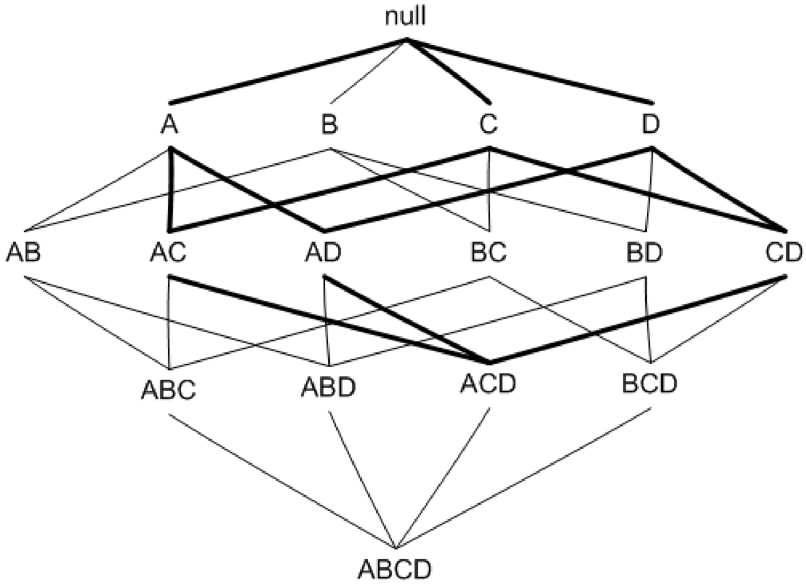


Figure 2.6: Large subsets of ABCD

A weakness of the Apriori algorithm is the number of database scans needed in the process. The maximum number of scans is one larger then than the cardinality of the largest large itemset. For this reason, some variants of Apriori has been proposed. A short description of two of these is given in the following.

Apriori with Transaction Reduction

This variant tries to reduce the number of transactions in the database [14]. If a transaction does not contain any large k-itemset, it can be removed from the database, or marked in someway, because of the Apriori-property. By doing this, the transaction will not be considered in subsequent scans of the database.

Apriori with Partitioning

Apriori with partitioning reduces the number of database scans by partitioning the database into n non-overlapping partitions [14]. The algorithm works by finding the large itemsets in each partition, these are called **local large itemsets**. The second phase consists of finding the actual support count for each of the local large itemsets to determine the global frequent itemsets.

2.4.2 FITI

The following presentation of the FITI-algorithm is adapted from [44]. The FITI algorithm is based on the following property, *A large intertransaction itemset must be made up of large intratransaction itemsets*, which means that for an itemset to be large in intertransaction association rule mining, it also has to be large using traditional intratransaction rule mining methods.

By using this property, the complexity of the mining process can be reduced, and mining intertransaction association rules can be performed in a reasonable amount of time. Before going into more detail about how the FITI algorithm works, an overview of some basic concepts will be given.

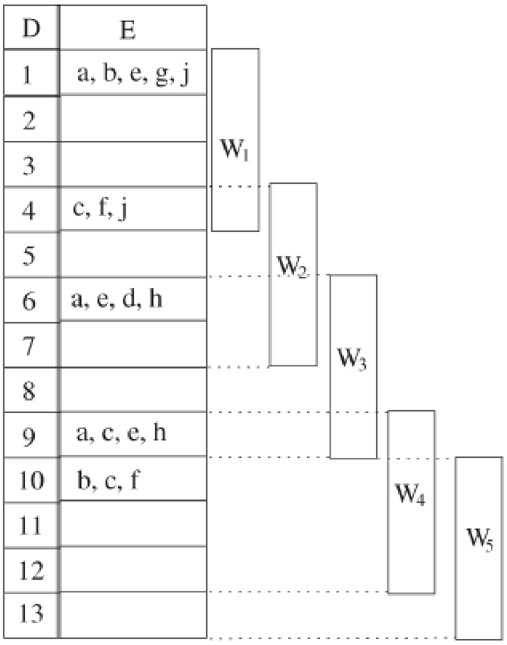
First, FITI introduces a parameter called **maxspan** (or sliding window size), denoted w. This parameter is used in the mining of association rules, and only rules spanning less than or equal to w transactions will be mined. The sliding window, denoted **W**, consists of w subwindows. Figure 2.7 shows a database with 5 transactions, located at intervals 1, 4, 6, 9 and 10, and 5 sliding windows, W_1 through W_5. Subwindow $W_2[0]$ contains the items {c, f, j}.

Second, every sliding window in the database forms a **megatransaction**. A megatransaction in a sliding window W is defined as the set of items W, appended with the subwindow number of each item. Using this on Figure 2.7, the megatransaction in W_3 becomes {a(0), e(0), d(0), h(0), a(3), c(3), e(3), h(3)}. The items in the megatransactions are called **extended items**.

Finally, the support and confidence values for intertransaction rules differ slightly from the values for intratransaction association rules. Let S be the number of transactions in the database. T_{xy} is the set of megatransactions that contain the set of extended items X∪Y, and T_x is the set of megatransactions that contain X. The support of an intertransaction association rule X ⇒ Y is then defined as:

$$support = \frac{|T_{xy}|}{S}, confidence = \frac{|T_{xy}|}{|T_x|}$$

Now, an overview of the FITI algorithm will be given. The algorithm consists of three phases, listed below.



D	E
1	a, b, e, g, j
2	
3	
4	c, f, j
5	
6	a, e, d, h
7	
8	
9	a, c, e, h
10	b, c, f
11	
12	
13	

Figure 2.7: Transactions and sliding windows

- **Mining and storing large intratransaction itemsets**
- **Database transformation**
- **Mining large intertransaction itemsets**

These phases are described in more detail in the following.

Mining and Storing Large Intratransaction Itemsets

The first phase of FITI is concerned with mining large intratransaction itemsets. For this task, any traditional intratransaction rule mining algorithm may be used, for example Apriori, described in Subsection 2.4.1. The large itemsets are stored in a structure called **Frequent-Itemsets Linked Table**, or simply **FILT**. This datastructure consists of an *ItemSet Hash Table*, and nodes are linked by several kinds of links. Figure 2.8 shows the **lookup links**. These link each large intratransaction itemset to its unique ID number, which corresponds to a row number in the Itemset Hash Table. For an overview of the other types of links, please see [44].

Database transformation

The second phase of FITI is to transform the database into a set of *encoded Frequent-Itemset Tables*, called **FIT** tables. There will be a total of $\mathbf{max}_k$ tables, where $\max_k$ is the maximum size of the intratransaction itemsets discovered in the first phase. Recall from Figure 2.8 that the largest itemset had a size of 3, the itemset {a, b, c}. This will therefore give three FIT tables, shown in Figure 2.9.

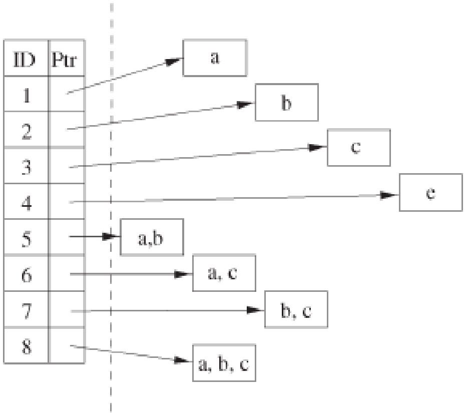
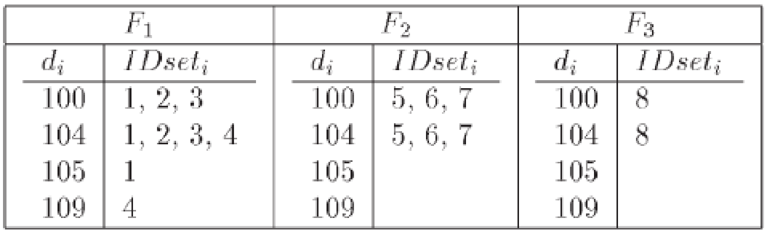


Figure 2.8: FILT structure (adapted from [44])



F_1		F_2		F_3	
d_i	$IDset_i$	d_i	$IDset_i$	d_i	$IDset_i$
100	1, 2, 3	100	5, 6, 7	100	8
104	1, 2, 3, 4	104	5, 6, 7	104	8
105	1	105		105	
109	4	109		109	

Figure 2.9: FIT tables (adapted from [44])

Each of the tables, F_k, will be on the form {d_i, $IDset_i$}, where d_i is the value of the dimensional attribute (in this case transaction number in the database), and $IDset_i$ is the IDs of the large k-itemsets in the FILT structure.

The FIT tables are organized so that table F_k contains the large k-itemsets. For example, the entries for F_2 are {100, [5, 6, 7]} and {104, [5, 6, 7]} which means that both transactions number 100 and 104 contain the 2-itemsets with IDs 5,6 and 7 in the FILT structure in Figure 2.8. For more details on the transformation, please see [44].

Mining Large Intertransaction Itemsets

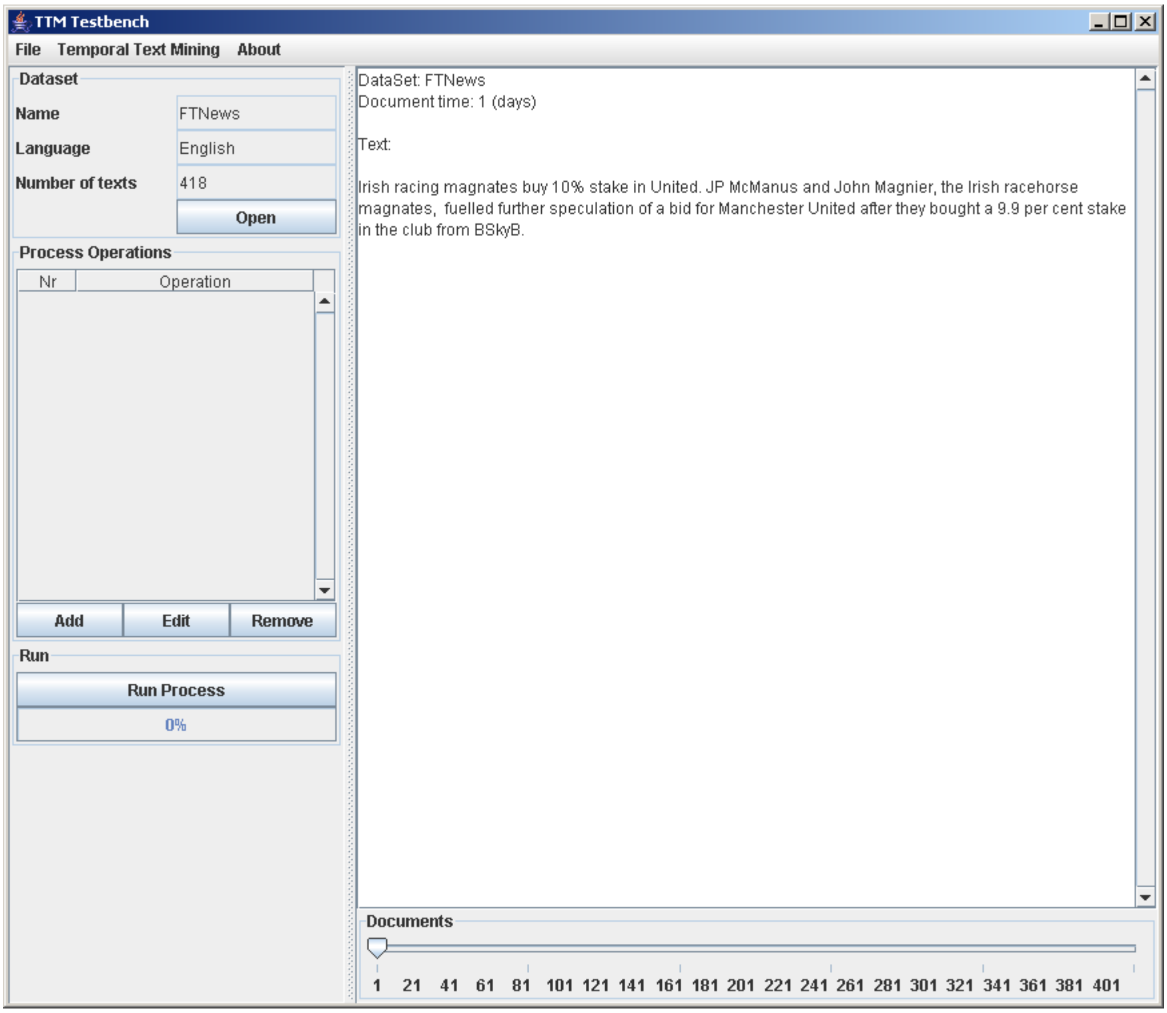
The final phase of FITI is the mining of large intertransaction itemsets. The mining algorithm in FITI proceeds in a level-wise fashion, this means using *k*-itemsets (for $k \geq 2$) to form candidate (*k+1*)-itemsets. For every candidate (*k+1*)-itemsets, the FIT-tables are checked to see if the corresponding intratransaction itemset is present. If it is not present, the candidate itemset is removed.

The algorithm runs on top of an input layer, which provides a sliding window. This helps optimize the algorithm in two ways. First, the input is limited to a set of essential FIT tables instead of reading the whole set. For example, when counting the support for candidate 2-itemsets, only F_1 needs to be accessed since all 2-itemsets are made up of large 1-itemsets. Second, as the sliding window is moved along, new transactions will be filtered to remove the IDs of large intratransaction itemsets which are not present in any candidate intertransaction itemsets.

This was just a brief introduction of the various phases of the FITI-algorithm, for a more detailed overview and examples, please see [44].

2.5 The Temporal Text Mining (TTM) Testbench

The TTM Testbench is an application developed for experimenting with various text preprocessing operations and rule mining algorithms on a temporal document collection. The application was initially developed by two other students in a project the fall of 2005. Further developments was introduced in the project the fall of 2006. Figure 2.10 shows a screenshot of the graphical user interface of the application. The TTM Testbench has been developed entirely in Java.



Name: FTNews
Language: English
Number of texts: 418

DataSet: FTNews
Document time: 1 (days)

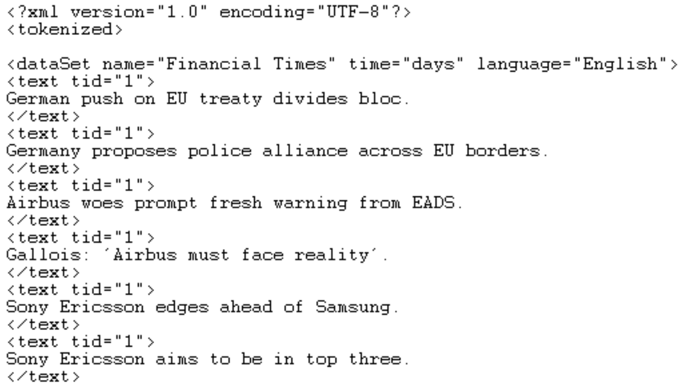
Text:

Irish racing magnates buy 10% stake in United. JP McManus and John Magnier, the Irish racehorse magnates, fuelled further speculation of a bid for Manchester United after they bought a 9.9 per cent stake in the club from BSkyB.

Figure 2.10: Screenshot of the TTM Testbench GUI

The TTM Testbench takes as input a specially prepared XML file containing the individual text items, these correspond to news items in this project. Each text item is marked with a timestamp indicating which document they are a member of. A part of this structure is shown in Figure 2.11.

After a document set has been loaded, the user proceeds by selecting the relevant text preprocessing operations and rule mining algorithm along with wanted parameters for each opera-



```
<?xml version="1.0" encoding="UTF-8"?>
<tokenized>

<dataSet name="Financial Times" time="days" language="English">
<text tid="1">
German push on EU treaty divides bloc.
</text>
<text tid="1">
Germany proposes police alliance across EU borders.
</text>
<text tid="1">
Airbus woes prompt fresh warning from EADS.
</text>
<text tid="1">
Gallois: 'Airbus must face reality'.
</text>
<text tid="1">
Sony Ericsson edges ahead of Samsung.
</text>
<text tid="1">
Sony Ericsson aims to be in top three.
</text>
```

Figure 2.11: Structure of the XML files taken as input in the TTM Testbench

tion. The TTM Testbench then executes the selected operations, and the output is association rules which span one or more versions of the document (analogous to transactions in a regular database).

The TTM Testbench has also been modified to run without the graphical user interface shown above. This was done to make it possible to run the Testbench on a remote server with more computing power than a regular desktop computer. This version of the Testbench reads a configuration file where the user specifies which document collection to use, which operations to perform, and the parameters of the operations.

2.5.1 TTM Testbench Preprocessing Operations

The main task of the text preprocessing operations is to prepare the text documents for further mining. This is done by trying to identify and extract the most relevant document features from each document.

The implemented operations are listed and described below.

- **Extract terms** - This operations simply extracts all words from the individual texts. All words are cleaned by removing non-letters and digits. The frequency of all words are stored.
- **Part of speech tagging (Introduced in 2006)** - This operation works in two phases. First, all words are tagged with their most likely part of speech tag (e.g. verb or noun). Second, words with specific, user-specified, tags are extracted. The user can specify whether he wants to extract common nouns, verbs, adjectives, numbers and adverbs. In addition, the operation extracts proper nouns, and various patterns of proper nouns, for example ”Wall Street” and ”Ministry of Defence”. Part of speech tagging is described further in Section 3.2.
- **Extract nouns** - This operation implements a simple method for detecting and extracting nouns from the texts. The method works by extracting all words which starts with a capital letter. This means that all words at the beginning of a sentence will also be extracted even though they are not necessarily nouns.
- **Remove stopwords** - Removes stopwords from the term list. A stopword is a common or general word, such as ”that”, ”is” or ”and”. This operation is vital in reducing the number

of terms extracted from the texts, and may reduce the time spent mining for association rules.

- **Stem words** - The stemming operation tries to reduce the words to their grammatical roots, the stem [2]. The operation implements the well known Porter Stemmer, which uses a suffix list and a list of rules to remove suffixes from the words. For example, ”Walking” becomes ”Walk”. This operation is useful in reducing in reducing the number of distinct terms, and potentially reduces the time spent in the mining process.
- **Weight terms** - This method implements the **Term Frequency - Inverse Document Frequency** measure. This measure tries to quantify the importance of an term in a text by taking into account how often it appears in the text, and how often it appears in whole the document collection [2]. TF-IDF is discussed further in Subsection 3.5.1.
- **Filter terms** - This operation makes it possible to filter the terms extracted by the previous operations. The user must specify how many terms should be kept from each text. If the weighting operation has been used, the terms may be kept based on their weight. Otherwise, the *n* first terms are kept.

2.5.2 TTM Testbench Rule Mining Algorithm

The algorithm implemented in the TTM Testbench is a slightly simplified version of the FITI algorithm, presented in Subsection 2.4.2. For example, the implementation does not use hashing as described in [44].

The user is able to specify a number of parameters for the rule mining algorithm. These are given below.

- **Minimum support** - The minimum threshold for the support of the rules.
- **Maximum support** - The maximum threshold for the support of the rules.
- **Minimum confidence** - The minimum threshold for the confidence of the rules.
- **Maximum confidence** - The maximum threshold for the confidence of the rules.
- **Maxspan** - The maximum number of transactions a rule may span.
- **Maximum set size** - The maximum number of items present in an association rule.

2.5.3 News Item Extractor

In the project last fall, an application was implemented to extract news items from a collection of web newspapers. This was done because the extraction method in the original Testbench was prone to noise and errors in the extraction phase. In addition, the original extraction method was specifically designed for extraction of news items from Financial Times.

The new extractor was implemented based on the work presented in [35]. The extractor works by looking for three specific patterns in the HTML files, and by using this new extractor much of the noise originally extracted was left out.

The news items found in the extraction process is stored in a XML file, with the structure shown in Figure 2.11, which can be loaded by the TTM Testbench.

2.5.4 Results and Discussion

This part will give a brief presentation of the results of the experimentation in the two earlier projects. The results from the first project can be seen in [34]. This project performed experimentation with various combinations of the operations listed above, except part of speech tagging. The rules found in this project were not considered meaningful or interesting, and the conclusion was that a more semantically based approach was needed. Examples of rules from this project were (back, 0) ⇒ (Iraq, 1), (John, 0) ⇒ (hit, 1), and ((UK, 0) (Bush, 1)) ⇒ (Iraq, 2).

In the project last fall, part of speech tagging was introduced. This was done to focus on semantics in the document feature extraction phase. The rules found can not be considered really interesting, and the words present in the rules are words which are common in a domain of financial news, for example ”business”, ”profits” and ”banks”. Examples of rules found are (bank, 0) ⇒ (business, 1), (economy, 0) ⇒ (bush, 1), and ((attack, 0) (profits, 1)) ⇒ (bush, 2). The full set of rules can be seen in C.1.

The document collection used in the experimentation was the same for both projects. This collection consists of 37 versions of the front page of the Financial Times[2]. This is considered a very small dataset when mining for knowledge. Future experiments should therefore seek to use much larger datasets.

In addition, the rules found in both projects can not be considered very interesting, and their meaning was not always clear. It might therefore be advantageous to move from single words to compound terms or concepts in the document feature extraction process. There are also some general challenges regarding working with textual data, these will be discussed in 3.1.

2.6 Summary

This chapter has provided an overview of data and text mining, in particular association rules, and the previous work on the TTM Testbench. In the next chapter, an analysis of various methods which has the potential to improve the results from the project last fall is given.

[2] http://www.ft.com/home/europe

Chapter 3

Analysis

This chapter will give an overview, and analysis, of fields of interests to this project. The chapter starts by presenting known challenges, these are issues which needs to be considered during analysis of the fields of interest. The chapter then continues with an analysis of the different fields of interest.

3.1 Known Challenges

Before going into detail about the fields of interest, this section will give an overview of some of the challenges which have to be kept in mind when the analysis of potential methods and techniques is performed. First, a presentation of challenges regarding text and natural language documents is given. Then, an overview of challenges with association rules is given, this includes both the suitability of association rules, and the problem of finding the most interesting rules.

3.1.1 Challenges with Textual Data

There are several challenges which come into play when textual documents are used in the mining process instead of structured databases. The main challenges are presented below.

Feature Dimensionality

A major challenge in text mining is the high feature dimensionality of textual documents, that is, the size and scale of possible combinations of the feature values in the dataset.

High feature dimensionality is typically a problem of greater magnitude in text mining systems than in data mining systems, since natural language documents have a larger number of potential representative document features, and thus combinations of feature values, than is generally found in relational or hierarchical databases [11].

It is therefore essential for a text mining system to include preprocessing operations which can help reduce the dimensionality, and create a streamlined representational model.

Feature Sparsity

Another challenge which appears when using textual documents in the mining process is something which can be described as feature sparsity [11]. This means that a single document only contains a small fraction of all possible document features in the document collection as a whole.

The implication of this is that when the documents are represented as a binary vector of features, almost every value of the vector is zero. That is, many of the features appear only in a small number of documents, and thus the support of many patterns is low.

Synonyms

Synonyms are words which have the same, or almost the same, meaning, and thus are interchangeable in a text. For example the words ”fabricate” and ”manufacture”.

The challenge of synonyms is closely related to both high feature dimensionality and feature sparsity. First, various synonyms for a word may be present in the document collection, each adding to the feature dimensionality. Second, many of the synonyms may be present only one or two times in the whole collection, and therefore have low support.

The preprocessing phase should therefore seek to identify synonyms in the text, and let all synonyms be represented by the same word. For example, both ”fabricate”, ”manufacture” and ”construct” are represented by the word ”construct”.

Homonyms and Polysemy

Homonyms are words with the same spelling (or pronunciation), but with different meanings, this property of a word is called **polysemy**. The word ”bank” for example has at least two different meanings. It can mean both a financial institution where people can deposit and withdraw money, and it can mean the slope beside a body of water.

Homonyms should be considered when designing a text mining system. This has to do with the meaningfulness of the results of the mining process. Consider for example the following association rule, $\{\text{bank} \Rightarrow \text{bush}\}$. Both terms in this association rule are polysemes, meaning that the words have several meanings. It is therefore difficult, if not impossible, for a user to fully understand the meaning of the association rule.

In addition, the words may appear with different meanings in the document collection, but the mining algorithm will consider them several instances of the same word. This can lead to association rules being returned by the rule mining algorithm because the combined support count is above the minimum support threshold even though each of the different meanings of the word may have too low support.

3.1.2 Association Rule Mining

There are some challenges regarding association rules as well. The first challenge relates to the suitability of association rules in this project, and the second challenge concerns itself with the problem of finding interesting rules.

Suitability of Association Rules

It is still unclear whether association rules are suited for discovering the temporal relationships mentioned in the problem definition. The two previous projects on the TTM Testbench have shown that it is possible to find relationships between words in different versions of a textual document. However, the rules can not be considered very interesting.

Further experiments will therefore have to be performed before any conclusions on the suitability of association rules can be made, especially experiments with more elaborate feature extraction methods which can identify and extract semantically rich document features.

Finding the Most Interesting Rules

A problem regarding association rules in general, is that the result set from the rule mining process usually contains a huge number of association rules. This is certainly the case in text mining, where the feature dimensionality may be very high, and thus the possible combinations of document features huge. It is therefore important to look at techniques and methods for filtering the result set, so that only the most interesting rules are presented.

3.2 Part of Speech Tagging

Part of speech tagging was introduced in the project last fall, but since part of speech tagged words are useful for some of the techniques and methods presented later in this chapter, a presentation of the subject will be given here as well. This presentation is an adaptation of the one given last fall.

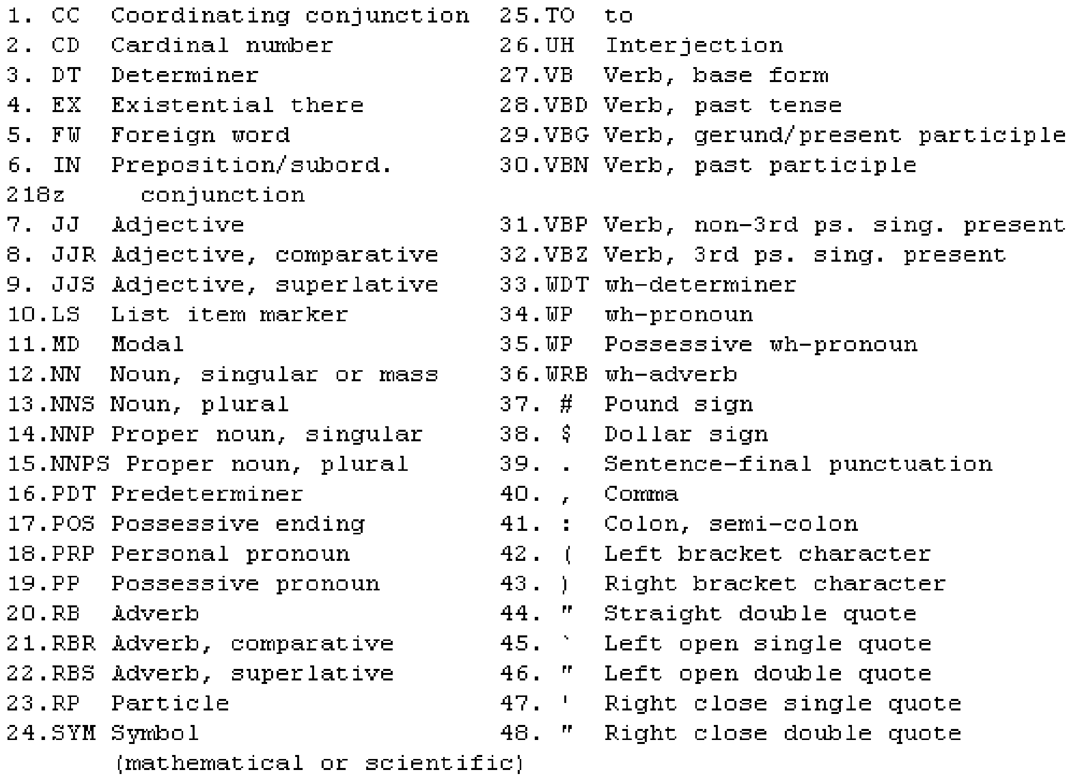
Part of speech tagging can be defined as the process of assigning a part of speech, or other lexical class marker, to each word in a corpus [18]. The part of speech tags divide the words in a sentence into categories based on the role they play in the sentence, and provides information about their semantic content [11].

The process of POS tagging can be seen as a disambiguation problem, where the tagger algorithm returns the most likely tag for a word based on the tags of the words in the surrounding context. Consider for example the word ”bank”, this can be used as both a noun, e.g. ”He went to the bank to deposit money”, and as a verb, e.g. ”The aircraft started to bank”. Table 3.1 gives an illustration of the ambiguity problem for the words in the Brown Corpus.

Table 3.1: Ambiguity of words in the Brown Corpus, adapted from [18]

Possible tags	**Number of words**
1 (unambiguous)	35340
2	3760
3	264
4	61
5	12
6	2
7	1 - The word ”still”

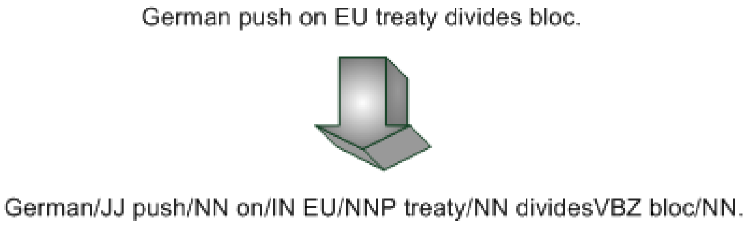
The tags returned for each word in the tagging process will depend on which set of tags (tagset) is being used. However, most tagsets use the same basic categories, and the most common set of tags include article, noun, verb, adjective, preposition, number and proper noun [11]. A well known tagset is the Penn Treebank tagset, and the different tags and their description is shown in Figure 3.1. Other tagsets include the C7 tagset and the UCREL CLAWS Tagset.



1. CC	Coordinating conjunction	25.TO	to
2. CD	Cardinal number	26.UH	Interjection
3. DT	Determiner	27.VB	Verb, base form
4. EX	Existential there	28.VBD	Verb, past tense
5. FW	Foreign word	29.VBG	Verb, gerund/present participle
6. IN	Preposition/subord.	30.VBN	Verb, past participle
218z	conjunction		
7. JJ	Adjective	31.VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32.VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33.WDT	wh-determiner
10.LS	List item marker	34.WP	wh-pronoun
11.MD	Modal	35.WP	Possessive wh-pronoun
12.NN	Noun, singular or mass	36.WRB	wh-adverb
13.NNS	Noun, plural	37. #	Pound sign
14.NNP	Proper noun, singular	38. $	Dollar sign
15.NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16.PDT	Predeterminer	40. ,	Comma
17.POS	Possessive ending	41. :	Colon, semi-colon
18.PRP	Personal pronoun	42. (	Left bracket character
19.PP	Possessive pronoun	43.)	Right bracket character
20.RB	Adverb	44. "	Straight double quote
21.RBR	Adverb, comparative	45. `	Left open single quote
22.RBS	Adverb, superlative	46. "	Left open double quote
23.RP	Particle	47. '	Right close single quote
24.SYM	Symbol (mathematical or scientific)	48. "	Right close double quote

Figure 3.1: The Penn Treebank tagset

Figure 3.2 shows the result of performing part of speech tagging on a sentence from a news article. The tagging was performed using the Penn Treebank tagset. This shows that it is possible to extract only words tagged with specific tags, for example nouns and proper nouns.



German push on EU treaty divides bloc.

German/JJ push/NN on/IN EU/NNP treaty/NN dividesVBZ bloc/NN.

Figure 3.2: Result of performing part-of-speech tagging

3.2.1 Part of Speech Tagger Classes

Most part of speech tagging algorithm are classified into one of two classes, **rule-based** taggers and **stochastic** taggers. In addition, some taggers include features from both these classes, such taggers are called **transformation-based** taggers. A short description of each of these three classes will be given in the following, for a more detailed presentation„ please see [18].

Rule-Based Part of Speech Tagging

Rule-based taggers are usually based on a two stage architecture [18]. First, a dictionary, or other source, is used to list all possible tags for a word. The second stage consists of applying a large set of rules or constraints to narrow down the list to a single part of speech tag. An example of such a rule is shown below.

> **Input word:** ”that”
> *If next word is adjective, adverb or quantifier, and following is a sentence boundary.*
> *And the previous word is not a verb like ”consider” which allows adjectives as object complements.*
> *Then eliminate non-adverb tags.*
> *Else eliminate adverb tag.*

An example of a rule-based tagger is the ENGTWOL tagger [45].

Stochastic Part of Speech Tagging

Stochastic taggers use probabilities when deciding the tag for each word in a given sentence [18]. An algorithm which uses stochastic tagging is the Hidden Markov Model (HMM) tagger. HMM taggers work by choosing the tag sequence which maximizes the following formula.

$$P(word|tag) \times P(tag|previous\, n\, tags) \tag{3.1}$$

The probabilities used in the calculation of the formula shown above, are collected from one or more manually tagged corpora, for example the Brown or the Switchboard corpora. For example, the likelihood for respectively a noun and a verb to appear after the word ”to” in the combined Brown and Switchboard corpora is shown below.

$$\begin{aligned} P(NN|TO) &= 0.021 \\ P(VB|TO) &= 0.34 \end{aligned}$$

Transformation-Based Part of Speech Tagging

Transformation-based tagging draws inspiration from both rule-based and stochastic tagging, and is an instance of the **Transformation-Based Learning** approach to machine learning [18]. This kind of tagging is sometimes called Brill tagging after the tagger presented in [7].

As with rule-based taggers, transformation-based taggers are based on the application of rules which specify what tag a word should be assigned. In addition, like stochastic tagging, transformation-based tagging is a machine learning technique. Transformation-based taggers induce the rules automatically from training data.

Transformation-based taggers work by applying increasingly narrow rules to the corpus. First, the corpus is tagged using the rule which applies to the most cases. Then, more specific rules are used, which changes some of the original tags.

3.2.2 Part of Speech Tagging in the TTM Testbench

As mentioned earlier, part of speech tagging is already implemented as a text preprocessing operation in the TTM Testbench. This operation uses the **Stanford Log-linear Part-Of-Speech Tagger**[1] to tag the document collection. This tagger uses a Maximum Entropy model, which is similar to stochastic tagging. Details about this tagger is presented in [43].

After the texts in the document collection are tagged, the operation extracts words tagged with one of a set of user-specified part of speech tags. This includes nouns, proper nouns and proper noun groups, verbs, adjectives, numbers and adverbs.

3.3 WordNet

As with part of speech tagging, WordNet[38] was also presented in the project last fall. But to keep this report self-contained, an updated and adapted presentation of WordNet will be given here as well.

WordNet was created at the Cognitive Science Laboratory at Princeton University, and is a lexical reference system inspired by psycholinguistic theories of human lexical memory [29]. The work on WordNet started at Princeton University in 1985, and it is still being maintained. The goal of WordNet was to provide an aid for searching dictionaries conceptually rather than alphabetically [29].

The main difference between WordNet and a regular dictionary is that the WordNet lexicon is divided into nouns, verbs, adjectives and adverbs. The drawback of this approach is that there is some redundancy, since many words appear in several categories. The advantage of the approach is that differences in the semantic organization of these syntactic categories can be seen and exploited. This will be illustrated in the following when a description of the organization of WordNet is given.

3.3.1 Organization of WordNet

WordNet is organized into synonym sets (synsets), where each synset corresponds to one underlying lexical concept. Each synset consists of one or more synonyms, and a gloss describing the concept it represents. For example, one of the synsets for the word *bank* is {*depository financial institution, bank, banking concern, banking company*} with the following gloss: *a financial institution that accepts deposits and channels the money into lending activities.* Many of the synsets in WordNet also contains examples of usage, for example "*he cashed a check at the bank*".

The various synsets in WordNet are linked together through semantic relations. These relations differ according to whether the synset represents a noun, verb or adjective. For nouns, the relations are *hypernyms, hyponyms, meronyms* and *holonyms.* A short description of these is given below.

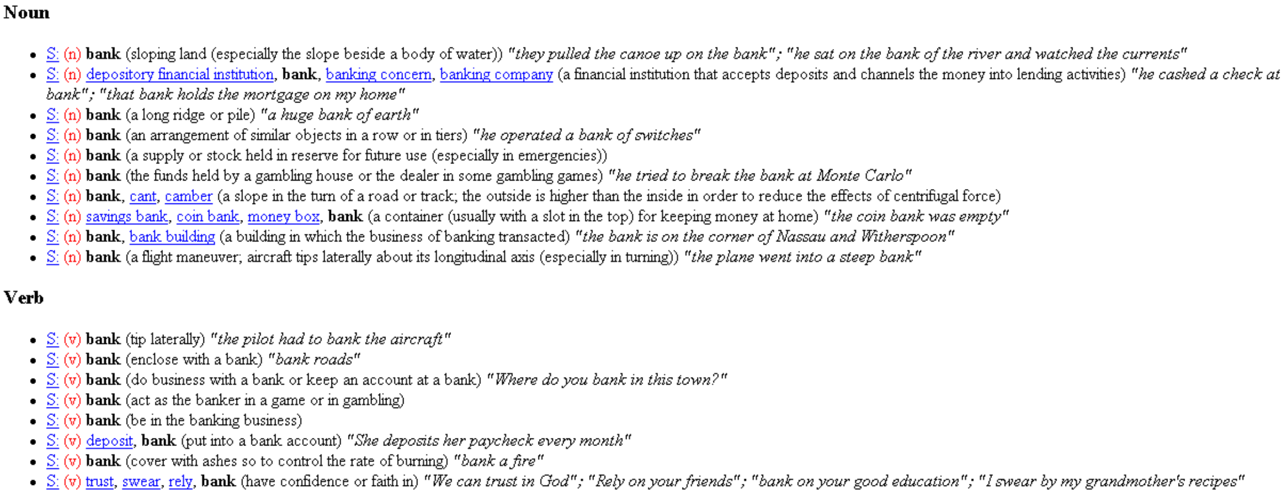
[1] http://nlp.stanford.edu/software/tagger.shtml

- Hypernym - A is a hypernym of B if B is-a-kind-of A, for example *car is a hypernym of convertible*
- Hyponym - B is a hyponym of A if B is-a-kind-of A, for example *mountain bike is a hyponym of bicycle*
- Meronym - B is a meronym of A if B is-a-part-of A, for example *air bag is a meronym of car*
- Holonym - A is a holonym of B if B is-a-part-of A, for example *bicycle is a holonym of pedal*

Verb synsets are also linked together in WordNet, the relations for verbs are *hypernyms* and *troponyms*. Hypernym relations for verbs are similar to those for nouns, a verb synset A is a hypernym of B if *to* B *is one way to* A, for example walk is hypernym of march. Troponyms are analogous to the hyponym relations for nouns, thus march is a troponym of walk.

Finally, the relation linking adjective synsets together is the *similar-to* relation. For example, beautiful is similar-to pretty.

Figure 3.3 shows the result of performing a search for the word ”bank” in the web interface of Wordnet[38]. As one can see from the figure there are 10 synsets for ”bank” as a noun and 8 synsets for ”bank” as a verb. These synsets show the different **senses** the word can have. The task of deciding which sense of a word is used in a text can be useful in both information retrieval and text mining, and this task, called word sense disambiguation, described in Subsection 3.4.



Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in emergencies))
- S: (n) **bank** (the funds held by a gambling house or the dealer in some gambling games) *"he tried to break the bank at Monte Carlo"*
- S: (n) **bank**, cant, camber (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)
- S: (n) savings bank, coin bank, money box, **bank** (a container (usually with a slot in the top) for keeping money at home) *"the coin bank was empty"*
- S: (n) **bank**, bank building (a building in which the business of banking transacted) *"the bank is on the corner of Nassau and Witherspoon"*
- S: (n) **bank** (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)) *"the plane went into a steep bank"*

Verb

- S: (v) **bank** (tip laterally) *"the pilot had to bank the aircraft"*
- S: (v) **bank** (enclose with a bank) *"bank roads"*
- S: (v) **bank** (do business with a bank or keep an account at a bank) *"Where do you bank in this town?"*
- S: (v) **bank** (act as the banker in a game or in gambling)
- S: (v) **bank** (be in the banking business)
- S: (v) deposit, **bank** (put into a bank account) *"She deposits her paycheck every month"*
- S: (v) **bank** (cover with ashes so to control the rate of burning) *"bank a fire"*
- S: (v) trust, swear, rely, **bank** (have confidence or faith in) *"We can trust in God"; "Rely on your friends"; "bank on your good education"; "I swear by my grandmother's recipes"*

Figure 3.3: Entry of ”bank” in WordNet

3.3.2 Usage of WordNet

WordNet can be utilized in several ways in this project. First, WordNet can be utilized in the word sense disambiguation process described in Section 3.4. And second, it can be used in the process of extracting meaningful document features from the document collection, presented in

Section 3.5. In addition, it is possible to use WordNet in the process of rating the association rules, discussed in Section 3.6.

3.4 Word Sense Disambiguation

This section will give an overview of word sense disambiguation, and how it may be useful for this project. It will also give an overview of different methods and algorithms for performing the disambiguation.

Word Sense Disambiguation(WSD) is a topic of both theoretical and practial interest, and the task of WSD is to examine word tokens in a text and specify exactly which sense of each word is being used [18]. As an example, consider the word *club*, and two of its distinct senses:

- Club - golf equipment used by a golfer to hit a golf ball.
- Club - a playing card in the minor suit that has one or more black trefoils on it.

The consider the following sentences.

- He brought his clubs to the golf course.
- He chose clubs as trumps.

For a human it is obvious which sense of ”club” is used in the two sentences, the problem is however how to create robust algorithms for computers to automatically perform this task. An overview of different of techniques and algorithms will be given later in this section.

The process of WSD usually consists of two steps, given below.

1. Find all possible senses for all the relevant words in a text.
2. Assign each word its correct sense.

The first step is straightforward. As shown in 3.3, WordNet contains a number of synsets for each word, where each synset is a possible sense of the given word. Therefore, the first step can be accomplished by retrieving the possible senses from WordNet. WordNet is only one option for finding the possible senses, any available machine-readable dictionary, or knowledge source, may be used. This report, will however focus on WordNet, since Java interfaces to the WordNet dictionary are readily available.

The second step of WSD is accomplished by relying on two major information sources [16]. The first is the context of the word to be disambiguated, this includes both information within the text, and extra-linguistic information about the text, for example the situation. The second source is an external knowledge source such as lexical or encyclopedic resources, or hand-devised knowledge sources. The task of these sources is to provide data useful for associating a word with a sense.

The disambiguation work can be classified into two categories. These are listed below.

- **Knowledge-driven WSD** - The context of the word to be disambiguated is matched with information from an external knowledge source (e.g. WordNet).
- **Data-driven or corpus-based WSD** - The context of the word to be disambiguated is matched with the context of previously disambiguated instances of the word derived from corpora.

The best match between the current context and one of the sources mentioned above can be found by using a variety of *association methods*. Some of these methods will be presented later in this section.

3.4.1 Applications of WSD

WSD has many potential applications, for example *machine translation*, *information retrieval and hypertext navigation* and *content and thematic analysis* [16].

An obvious benefit of performing WSD in this project is to deal with the challenge of homonyms, presented in 3.1. By using WSD in the preprocessing phase, the document features presented to the rule mining algorithm may be tagged with their sense number, or some other marker, to differentiate equal words appearing with different senses.

An implication this may be that many of the rules found when mining without taking into account the sense of a word may be filtered out because the combined support count of the senses of the word may be above the minimum support threshold even though each of the different senses has a too low support. The opposite effect may also be found, the support count of several of the different senses may exceed the minimum support threshold, and the result may be a larger number of rules returned from the rule mining process.

Another benefit is that the terms may be expanded with synonyms, or even a description of the sense, to make the rules more meaningful. These synonyms and descriptions can be found by looking up the correct synset in WordNet.

Another useful aspect of WSD is that it becomes possible to use WordNet as an hierarchical knowledge structure for use in the feature extraction process. If the sense of a word is known, it is possible to extract more general or more specified terms, and it will also be possible to detect that two or more words are instances of some higher-level concept. This will be discussed in more detail in Subsection 3.5.4.

3.4.2 The Lesk Algorithm

This algorithm tries to automatically decide the sense of a word by using machine readable dictionaries to look for overlaps in the word sense definitions of the word being disambiguated and the description of the words in its context[23].

The method of using the context of a word for automatic sense disambiguation was first presented by Weaver in his famous *Memorandum*(1949) [16]. In this, Weaver writes the following:

> *If one examines the words in a book, one at a time through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of words. "Fast" may mean "rapid"; or it may mean "motionless"; and there is no way of telling which.*
>
> *But, if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then, if N is large enough one can unambiguously decide the meaning...*

As an overview of how this algorithm works, consider the following simplified example. Given the word *bank* in the context *river bank*, the algorithm works as follows. First, the definitions of the various senses of *bank* are looked up in a dictionary. By using WordNet, the following senses and descriptions for the word ”bank” can be found. Note that examples of usage is left out for all the senses.

1. bank - sloping land (especially the slope beside a body of water)
2. bank - a financial institution that accepts deposits and channels the money into lending activities
3. bank - a long ridge or pile
4. bank - an arrangement of similar objects in a row or in tiers
5. bank - a supply or stock held in reserve for future use (especially in emergencies)
6. bank - the funds held by a gambling house or the dealer in some gambling games
7. bank - a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force
8. bank - a container (usually with a slot in the top) for keeping money at home
9. bank - a building in which the business of banking transacted
10. bank - a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)

The next step is to compare the senses and descriptions of the word in question with senses and descriptions of the context words. WordNet lists only one sense for the word ”river”, shown below.

1. river - a large natural stream of water (larger than a creek)

By comparing the different senses for ”bank” and ”river”, the Lesk algorithm will correctly discover that the first sense of bank is the one used for ”bank” in this context since ”water” is present in the description of both ”bank”-sense1 and ”river”.

This was, of course, a simplified example. In a real world application the context will usually consist of more than one word. When more than one word is being used as the context, the number of overlaps between each of the glosses of the context words and the word to be disambiguated is computed. The sense with the highest number of overlaps is then returned as the result.

A challenge with this algorithm is to select how many of the words in the context should be used in the disambiguation. The more context words that are used, the more likely it is that the correct sense is returned, but at a higher computational cost than with fewer context words.

3.4.3 The Adapted Lesk Algorithm

This algorithm is an adaption of the Lesk algorithm presented above, and tries to mitigate the limitations of that algorithm. The main challenge with the original Lesk algorithm is that dictionary glosses tend to be fairly short, and may thus provide an insufficient vocabulary for fine-grained distinctions in relatedness [3]. Take WordNet for example, the average length of a gloss is just seven words.

A measure called *extended gloss overlaps* is introduced to overcome the problem of too short glosses. This measure expands the glosses of the word being disambiguated with the glosses of related words. Related words can be found in WordNet by using the relations shown in Subsection 3.3.1.

Figure 3.4 shows some of the words related to ”bank” in the sense of a financial depository institution. The related words shown is the hypernym of ”bank”, and four of its hyponyms. These words will then form an extended set of words, and their glosses are combined to form the **extended gloss** of ”bank”.

The same is done with all the words in the context, and the extended gloss of the word to be disambiguated is compared to the extended gloss of all the words in the context. As with the original Lesk algorithm, the sense which has the highest number of matches with the context words is returned as the result.

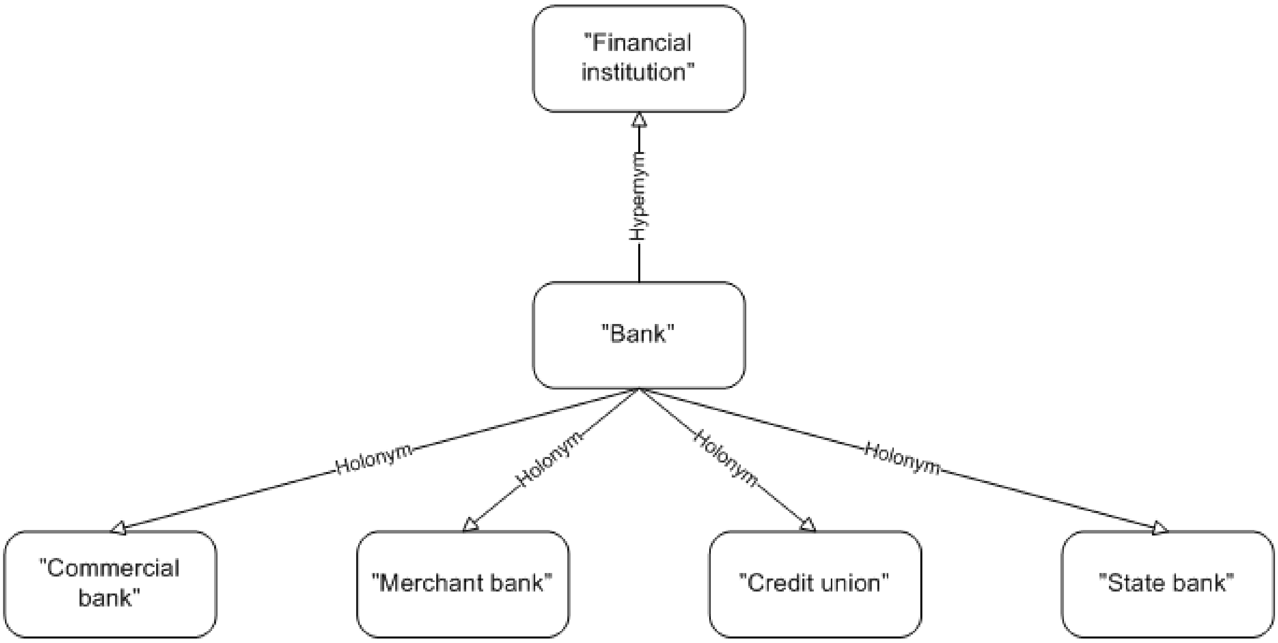


Figure 3.4: Bank (financial sense) and some of its related words

3.4.4 Other Algorithms and Methods

The two methods presented above can be classified as knowledge-driven WSD, there exists a number of other WSD methods which also use an external knowledge source. For an overview of some of these, please see [37]. An overview of approaches within knowledge-driven WSD is also given in [16].

There has also been a lot of work within corpus- and data-based WSD, both supervised and unsupervised methods. This includes for example training a **naive Bayes Classifier** or **decision list classifier** to perform the disambiguation. For more on corpus- and data-based WSD, please see [16] or [18]

3.5 Document Feature Extraction

This section will focus on the identification and extraction of descriptive and meaningful document features from the document collection. In information retrieval, this process is sometimes called *index term selection.* The reason for performing feature extraction is that using the set of all words in a document collection to represent its documents generate too much noise [2]. This is especially important when mining for association rules, where the user must be able to make sense of the rules in the result set, and the computational cost of using all words is very high.

There are various methods for feature extraction from a textual document. This report will focus on methods which has the potential to find semantically rich features. First, a presentation of TF-IDF will be given, before presenting methods with an increasing amount of semantic expressiveness.

3.5.1 Term Frequency - Inverse Document Frequency (TF-IDF) Measure

The TF-IDF measure is a method much used for weighting terms in information retrieval. It can be used both to find the potentially most meaningful document features among the native document features in a document, or to filter the document features found by using another method.

TF-IDF works by assigning a weight to each term, or document feature, in the document collection. Then, either the n highest ranked terms or all terms above some minimum weight threshold may be kept.

The TF-IDF measure consists of two separate measures. The TF part quantifies the frequency of term inside a document, while the IDF part quantifies the inverse frequency of a term among all documents in the collection. The last part thus dampens the weight of terms appearing in many documents.

The TF-IDF measure is calculated the following way.

Given:
N = the number of documents in the collection
n_i = the number of documents where term k_i occurs
$freq_{i,j}$ = the frequency of term k_i in document d_j
$max_l freq_{l,j}$ = the number of times the most frequent term is present in document d_j

The normalized frequency $f_{i,j}$ of term k_i in document d_j is given by:

$$\mathbf{TF_{i,j}} = \frac{\mathbf{freq_{i,j}}}{\mathbf{max_l freq_{l,j}}} \tag{3.2}$$

The inverse document frequency for term k_i is given by:

$$\mathbf{IDF_{i,j}} = \log \frac{\mathbf{N}}{\mathbf{n_i}} \tag{3.3}$$

Then, the weight of term k_i in document d_j is:

$$\mathbf{w_{i,j}} = \mathbf{TF_{i,j}} \times \mathbf{IDF_{i,j}} \tag{3.4}$$

3.5.2 Collocations

A collocation is an expression consisting of two or more words that correspond to some conventional way of saying things [27], for example ”weapon of mass destruction” or ”car bomb”. Collocations are common in natural languages, both in technical and non-technical texts. A word can not be classified only on the basis of its meaning, sometimes co-occurrence with other words may alter the meaning dramatically [5]. Consider for example ”to kick the bucket”, which actually means to die.

A characteristic of collocations is that they have limited **compositionality**, an expression is compositional if the meaning of the expression can be induced from the meaning of the parts. In addition, collocations can be said to have limited substitutability and limited modifiability [5]. This means that it is not possible to substitute any part of the collocation with a similar word, for example change from ”white wine” to ”yellow wine”. It is not possible to modify a collocation either, such as for example change from singular to plural form.

Collocations can be used for many purposes, and especially relevant to this project is word sense disambiguation and document feature extraction. For instance, if the words ”interest” and ”rate” appear next to each other in a text, and this is detected as the collocation ”interest rate”, this collocation can be used in the disambiguation process and will potentially improve the results. In addition, ”interest rate” is more meaningful than ”interest” and ”rate” by themselves, so a rule containing ”interest rate” may make more sense for a user than a rule containing only one of the words.

There are various statistical methods for finding collocations, a short description of two of these will be given in the following.

Frequency Counting

The simplest approach for finding collocations involves counting frequencies. The rationale behind this method is that words that tend to appear together, may have a meaning which is not simply explained as the result of their combination. The problem with this method is that it tends to find a lot of combinations of function words, such as ”of the”, ”in the” or ”is a”.

An improvement to this method was proposed by Justeson and Katz in [19], and uses an heuristic to filter out phrases which are not likely to be collocations. This heuristic consists of a part-of-speech filter which only keeps candidates with specific part-of-speech tags. Possible tag patterns can for example be noun-noun, e.g. ”oil price”, or adjective-noun, e.g. ”chief executive”.

Pearson’s Chi-square Test (Hypothesis Testing)

A shortcoming of frequency counting is that high frequencies can be accidental. To assess whether something happens more often than by chance, hypothesis testing can be used. This report will only discuss the chi-square ($\mathcal{X}^2$) test, but the T-test may also be used.

Hypothesis testing for collocations starts by formulating a *null hypothesis*, H_0, which states that the words occur together by chance. It is assumed that the two words, w_1 and w_2, are completely independent of each other. The probability of w_1 and w_2 occurring together is then:

$$\mathbf{P}(\mathbf{w_1}, \mathbf{w_2}) = \mathbf{P}(\mathbf{w_1})\mathbf{P}(\mathbf{w_2}) \tag{3.5}$$

The null hypothesis is then tested using statistical tests, for example chi-square or the *t*-test.

The essence of chi-square is that it compares the observed frequencies for the words with the frequencies expected for independence. Table 3.2 shows example values for the candidate collocation "old car" in a document collection.

Table 3.2: Examples of frequencies for chi-square calculation

	w$_1$ = old	$w_1 \neq$ old
w$_2$ = car	8 (old car)	4667 (e.g. new car)
w$_2 \neq$ car	15820 (e.g. old man)	142871181 (e.g. new company)

The chi-square formula is shown below.

Given:
N = the total number of tokens in the document collection
$O_{i,j}$ = the observed value for cell (i,j)
$E_{i,j}$ = the expected value

$$\mathcal{X}^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{3.6}$$

Chi-square can be calculated for for tables of any size, but for 2x2 tables it has a simpler form, shown below.

$$\mathcal{X}^2 = \frac{N(O_{1,1}O_{2,2} - O_{1,2}O_{2,1})^2}{(O_{1,1} + O_{1,2})(O_{1,1} + O_{2,1})(O_{1,2} + O_{2,2})(O_{2,1} + O_{2,2})} \tag{3.7}$$

The result of the calculation is then compared to the significance value for $\mathcal{X}^2$ with n degrees of freedom and a probability level of α. The calculation for the values in the table above is shown below.

$$\mathcal{X}^2 = \frac{14307668(8 * 142871181 - 4667 * 15820*)^2}{(8 + 4667)(8 + 15820)(4667 + 142871181)(15820 + 142871181)} \approx 1.55 \tag{3.8}$$

This example has 1 degree of freedom, and with a probability level of α=0.05 the critical value is 3.84. When comparing the result of the calculation with the critical value, one can see that the $\mathcal{X}^2$ value is below the critical value. The conclusion is therefore that "old car" is not a good candidate for a collocation.

3.5.3 Information Extraction

Information extraction (IE) is the task of locating specific pieces of data in natural language documents, thus extracting structured information from unstructured text [31], [30]. This makes it possible to move from a weakly structured textual format to a format which more closely resembles that of a structured database.

At present, there are four specific types of data which can be extracted from natural text [11].

- Entities - These are the basic building blocks in text. For example people, companies and locations.

- Attributes - These are features of entities. For example the title and age of a person.
- Facts - These are relations between entities. For example a person is employed in a company.
- Events - These are activities of interest which includes entities. For example a merger between two companies or the birthday of a person.

One particular type of IE which could be useful in this project is *named entity recognition*(NER). This type of IE involves identifying references to entities in the data, such as names of people, companies and locations [36], [30]. The NER phase of a text mining system can be said to be weakly domain dependent, and performance will depend on the similarity of the domain used while developing the NER engine and the domain of the documents [11].

An example of document features which could be extracted using NER is shown in Figure 3.5. As can be seen in the figure, using NER for extracting document features may lead to meaningful features being extracted. However, the features extracted are similar to those extracted when using the part-of-speech operation implemented in the TTM Testbench last fall. This operation extracts groups in proper nouns in a text, and can therefore, given that the words are tagged correctly by the POS-tagger, find both the document features shown in Figure 3.5, namely ”George W. Bush” and ”United States”.

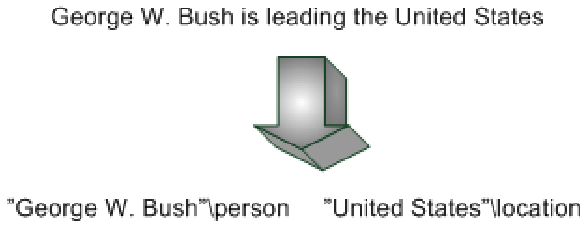


Figure 3.5: Example of result when using NER to extract document features

This was confirmed by preliminary experiments, where the Stanford named entity recognizer[2] was tested on parts of the original document collection of news articles from the Financial Times. The document features extracted when using NER were quite similar to those found when using the operation implementing part-of-speech tagging.

It was therefore decided that further developments should focus on using the already implemented operation, since part-of-speech tags are useful for other purposes as well, such as finding nouns to be candidates for word sense disambiguation.

3.5.4 Concept Extraction

The document feature extraction methods mentioned above, have all focused on finding document features which are native to the individual documents. There are however cases where it could be useful to extract higher-lever concepts or topics, this is especially the case when mining for association rules where support plays such a big role in the mining process.

It could for example be the case that several news stories are concerned with the topic ”vehicles”, but since different words like ”car”, ”automobile” or ”auto” may be used, no association rules are

[2] http://nlp.stanford.edu/software/CRF-NER.shtml

returned from the mining process. The following will give a brief overview of some methods for discovering topics or concepts in textual documents.

WordNet as an Ontology

An ontology is defined in [8] as *a formal, explicit specification of a shared conceptualization*, which is a combination of the definitions put forth in [13] and [6]. As the definition shows, an ontology has three characteristics, it is formal, explicit and shared. This means that an ontology is machine readable, the concepts of the ontology and the constraints are defined, and it captures consensual knowledge.

All these properties can be said to apply for WordNet. By utilizing the relationships in the hierarchical structure of WordNet it is possible to find higher-level concepts or topics. The most relevant relationship may be the **hypernym**-relation which models ”is-kind-of”-relationships between concepts in the hierarchy. The following relationship is for example part of WordNet, ”motor vehicle” is a hypernym of ”car”, which means that a car is a kind of motor vehicle.

In addition, WordNet contains the relationship ”category member” which is used to show that a concept is part of a higher-level category. An example of this is that ”lawsuit” is part of the category ”law, jurisprudence”, and ”troop movement” is part of the category ”military, armed forces”.

Both the uses shown above can be useful when trying to identify and extract concept-level document features in the document collection. An issue of using this method is whether to extract the original words as well as the concepts or categories which are found. Another issue is that to use this method, the words have to be disambiguated first. This is because the different senses of a word may be related to different higher-level concepts or categories.

Concept Counting

This technique, presented in [24], can be used to discover appropriate topics or concepts in textual documents. The technique is based on a knowledge-based concept counting paradigm, and tries to move beyond simple word counting methods, such as the TF-IDF measure.

Consider the sentence *”Michael bought some vegetables, fruit, bread and butter”*. By using a word counting method, like TF-IDF, one can draw no conclusion on what the higher-level topic or concept of this sentence is. This is because word counting methods fails to discover the concepts behind the words. Vegetables, fruit, etc. can all be said to relate to for example *groceries* or *food products* at a deeper level of semantics.

To overcome the limitation mentioned above, a method for identifying topics by counting concepts was introduced. The method makes use of a concept generalization taxonomy, such as WordNet. Figure 3.6 shows a possible hierarchy for *digital computer*. By using this hierarchy, if we find the words *desktop computer* and *portable computer* in a text, we can infer that (one of) the topic(s) of text is *personal computers*. In addition, if the text also contains the words *mainframe* and *workstation*, the topic is most likely related to *digital computer*.

An important issue regarding this method is how to find the most appropriate generalization. To overcome this problem, two measures called *concept frequency ratio* and *starting depth* are introduced. The weight of a concept C is the frequency of occurrence of the concept C and it’s

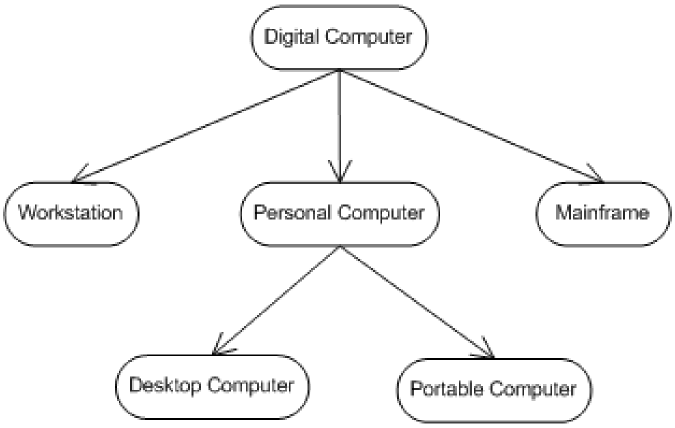


Figure 3.6: Sample hierarchy for *digital computer*

subconcepts. Then, the ratio R at any concept C is defined as:

$$\mathcal{R} = \frac{MAX(weight\,of\,all\,the\,direct\,children\,of\,C)}{SUM(weight\,of\,all\,the\,direct\,children\,of\,C)}$$

Recall the sentence ”Michael bought some vegetables, fruit, bread and butter.”. Assuming the words vegetables, fruit, bread and butter are all children of the concept ”food products”, the concept frequency ratio of ”food products” is then:

$$\mathcal{R}(food\,products) = \frac{1}{4} = 0.25$$

The higher the ratio, the less concept C generalizes over many children. This leads to the definition of the *branch ratio threshold* R_t:

> If a concept’s ratio R is less than R_t, it is an interesting concept.

If the branch ratio threshold of the example given above is over 0.25, the concept ”food products” would be identified as a suitable concept for representing the words in the sentence.

Domain Specific Taxonomy

It is also possible to simplify the problem by using a domain specific taxonomy. By doing this, the step of word sense disambiguation may be left out of the process since a domain specific structure most likely contains only the sense of a word which belongs to the specific domain. In this project, in which the previous work has performed the mining of association rules on news articles from a financial newspaper, a taxonomy or ontology dealing with the financial or business domain could be used.

Other

In [41], a method for mining generalized association rules are presented. This method uses an is-a hierarchy to find parents of the items in the transactions, which leads to the possibility of finding association rules with items from any level of the hierarchy.

However, this method finds association rules between items in the same transaction. No more consideration will therefore be given to this method. For more details on the method, please see [41].

3.6 Ranking of Association Rules

One of the problems with association rule mining is that the number of rules found may be huge. The user must therefore spend much time analyzing the rules to find the ones that are really interesting.

The goal of this section is to identify measures and methods for finding the potentially most interesting association rules, so that only these are presented to the user. Measures for rating association rules can be divided into two classes, *objective* and *subjective* measures [14]. Objective measures rate the association rules based on their structure, and the underlying statistical distribution of the data. Subjective measures rate the rules based on the users background, or his belief of the data. This can mean that the association rules are interesting if they are unexpected or novel.

The rest of this section will give a presentation of some of the possible measures for ranking association rules. Support and confidence were discussed in Subsection 2.3.1, so these measures are left out of this section. In addition to support and confidence, there is a number of other objective(statistical) measures. For an overview of some of these, please see [21].

3.6.1 Semantic Similarity

Words present in an association rule, which are close together(semantically related) in a knowledge hierarchy like WordNet, are more likely to be known by the user already. Therefore, rules where the words are less semantically related, can be considered more interesting [4].

The JCn Measure

This part will give a short introduction to one of these measures, presented in [17]. This measure is based on information content, and a short description of this will therefore be given. The information content of a concept is defined in [39] as the *negative log likelihood of encountering an instance of the concept.* The probability of a concept, c, is derived by dividing the frequency of the concept, freq(c), in a corpus by the number of concepts in the corpus, N. This formula is shown below.

$$IC(c) = -log(\frac{freq(c)}{N})$$

In addition, the measure also includes a distance approach to calculate the similarity between two concepts. The distance between two concepts in WordNet is the sum of the edge weights along the shortest path linking the two concepts [17]. The weight of a node is determined by factors such as local density in the hierarchy, node depth, and link (relation) type. For more information on this, please see [17].

The similarity measure of two concepts, c_1 and c_2, is then defined by the following formula, where c is the most specific concept in common between c_1 and c_2.

$$sim(c_1, c_2) = IC(c_1) + IC(c_2) - 2 * IC(c)$$

As an example, consider Figure 3.6. There, it can be seen that the most specific concept in common between ”desktop computer” and ”portable computer” is ”personal computer”.

3.6.2 User Knowledge

Interestingness is highly subjective. It may therefore be useful to somehow take into account the background or knowledge of the user, or a set of words or concepts the user may be interested in. This can for example be done by letting the user specify a list of terms which must be present in an association rule for it to be interesting, or a list of terms which are not interesting.

In addition, it could be possible for a user to store association rules which he has already seen in a knowledge base of some sort, and subsequent runs of the rule mining algorithm would only return rules not present in this knowledge base.

3.7 Summary

This chapter has given an overview of fields of interest to this project. This includes methods for extracting meaningful document features, and methods and measures for rating the potential interestingness of association rules. The next chapter will look at the improvements made to the TTM Testbench in this project.

Chapter 4

Temporal Text Mining Testbench

This chapter presents will present the improvements to the TTM Testbench. This includes both the gathering of new document collections, implementation of new operations in the TTM Testbench prototype, and modifications of existing operations. The chapter will start with an overview of the process of gathering a new document collection, before moving on to describe the improvements and changes made to the TTM Testbench application.

4.1 Document Collection

In previous work on the TTM Testbench, a document collection consisting of only 37 documents was used. This was considered too small to be able to find interesting association rules, and the need for a larger document collection was pointed out. The gathering of a new document collection was therefore of high priority in this project.

This section will first give an overview of how the new document collection was created. Then a description of how the news item extractor was modified to be able to handle the new collections will be given.

4.1.1 The Gathering Process

The new document collection was gathered using the UNIX-commands crontab[1] and GNU Wget[2]. A short description of these two commands is given below.

- **Crontab** - used to schedule periodical execution of commands. The period is specified by the user, and can for example be weekly, daily, every minute or every hour.
- **GNU Wget** - a command for retrieving content from web servers. The name is derived from the HTTP command GET.

By using crontab to periodically execute a Wget command specifying which document to retrieve, and where to store it, it was possible to generate a new document collection. In this project, three different crontabs were set up. The results of this is discussed in the following.

[1] http://www.opengroup.org/onlinepubs/009695399/utilities/crontab.html

[2] http://www.gnu.org/software/wget/

4.1.2 Overview of the new Document Collections

The first and second crontabs were similar. Both were set up to retrieve the front page of Financial Times[3], but at varying intervals. The first crontab was executed every 8 hours, while the second executed the Wget command every 12 hours. The results were two new document collections, one containing about 300 versions of the front page and the other containing about 200 versions.

The third crontab was set up to execute a Wget command to retrieve the front page of BBC Business[4]. This was done with a interval of 12 hours. This resulted in about 160 versions of the frontpage of BBC Business.

An overview of the three document collections is summarized in Table 4.1.

Table 4.1: Overview of the new document collections

Source	Number of days	Interval	Number of documents
Financial Times	About 100	Every 8 hours	About 300
Financial Times	About 100	Every 12 hours	About 200
Financial Times	About 80	Every 12 hours	About 160

One or more of these new document collections will be used for experimentation to see whether any improvements in the results compared to the original document collection can be seen.

4.1.3 Modification of the News Item Extractor

After the new document collections had been gathered, some issues regarding the news item extractor came to view during the extraction of news items from the different collections.

The first issue was performance. The new document collections had about 5-10 times as many documents as the original collection, and it became clear that the time spent extracting news items was too high. This was identified as a problem with the way the HTML files was read from the disk. The problem was solved, and the time used by the extractor was greatly reduced.

Another issue was that Financial Times and BBC Business used different character encodings in the HTML files, respectively ”UTF-8” and ”ISO-8859-1”. This caused problems when loading the created XML files into the TTM Testbench since the original extractor set the character encoding to ”ISO-8859-1”. The extractor was modified to detect which character encoding was used, and write the news items to the XML file using the appropriate encoding, in addition the correct encoding was added to the XML file, so that the TTM Testbench was able to load it correctly.

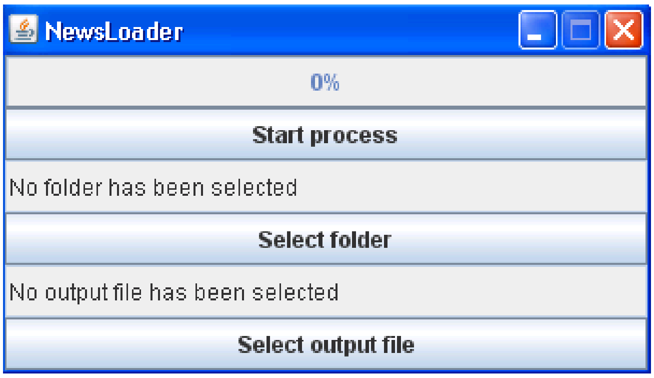
The final issue which was discovered, was that a lot of noise was extracted when the extractor was run on the new document collections. This was solved by making the extractor only look for news items in defined sections of the HTML files. The sections were selected based on manual inspection of the HTML files.

In addition, the original news item extractor used a configuration file where the user had to specify the location of the document collection, and where to store the results. This was removed because having to modify the configuration file proved very cumbersome. Both selection of the

[3] http://www.ft.com/home/europe

[4] http://news.bbc.co.uk/2/hi/business/default.stm

location of the document collection, and where to store the results, is now done in the graphical user interface (GUI). Figure 4.1 shows the new GUI of the news item extractor.



No folder has been selected

No output file has been selected

Figure 4.1: GUI of the new news item extractor

4.2 TTM Testbench

This section will give an overview of the changes to the TTM Testbench application, and the new operations introduced. It will start with an overview of the modifications, before describing the new operations.

4.2.1 Modifications of the TTM Testbench

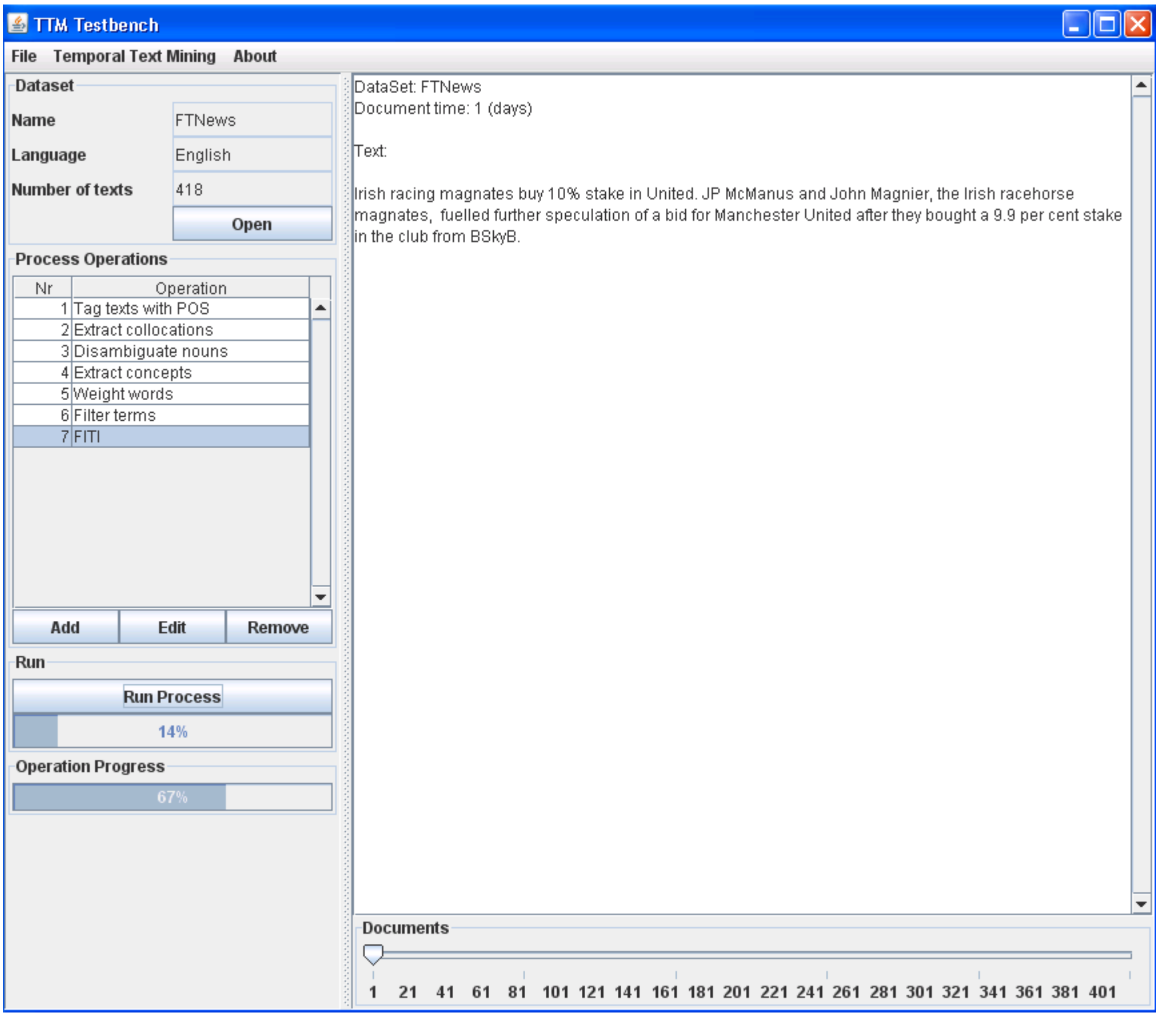
There has been made some changes to the original application. Both because new operations have been implemented and because some operations did not seem to give good results.

In addition, a progress bar was added to the GUI of the application. This bar indicates the progress of the operation currently being performed. This was done to increase the usability of the application, since some of the operations may take several minutes to complete, and the user is now able to see how far the operation has progressed. The new progress bar, labeled ”Operation Progress”, can be seen in Figure 4.2.

The Part of Speech Tagging Operation

In the original POS-operation, the texts were first tagged using the Stanford POS tagger. Terms were then extracted based on a set of word classes specified by the user. This operation has now been slightly altered to suit the collocation extraction method, presented in Subsection 4.2.2. Instead of extracting only specific word classes, all words are now extracted, and passed on to the collocation operation. In addition, the complete part of speech tagged text is sent along.

The filtering of terms based on their word classes is now done later in the process, this will be described in Subsection 4.2.2, which gives a description of the new operations.



Dataset

Name: FTNews

Language: English

Number of texts: 418

Process Operations

Nr	Operation
1	Tag texts with POS
2	Extract collocations
3	Disambiguate nouns
4	Extract concepts
5	Weight words
6	Filter terms
7	FITI

Run

14%

Operation Progress

67%

DataSet: FTNews
Document time: 1 (days)

Text:

Irish racing magnates buy 10% stake in United. JP McManus and John Magnier, the Irish racehorse magnates, fuelled further speculation of a bid for Manchester United after they bought a 9.9 per cent stake in the club from BSkyB.

Figure 4.2: Screenshot of the modified TTM Testbench GUI

Clustering for Filtering Association Rules

The original application included the possibility to filter association rules based on clustering. This was done by first clustering the news items using an external tool. Then, after the texts had been loaded into the Testbench, a file containing the cluster number for each text was read, and all terms in the text were marked with the cluster number of the text.

After the association rules had been mined, only rules which had terms from different clusters in the antecedent and the consequent were presented to the user. This is similar to the semantic similarity approach shown in 3.6.1, if one considers terms from the same cluster to be semantically similar.

However, this feature did not seem to work particularly well. One reason for this may be that some of the texts are very small, i.e. consisting of just one sentence. This may make it harder for the clustering algorithm to assign it to the correct cluster. The method has therefore been removed, and a new rule rating method has been implemented. The new method will be discussed in Subsection 4.2.3.

4.2.2 New Text Operations

This part of the report will give an overview of the new operations implemented in the TTM Testbench. First, an overview of the Java WordNet Library(JWNL) will be given. This is not an operation in itself, but a Java library used by the other operations to access the WordNet data. Following this, the new operations for processing the documents and extracting meaningful document features will be discussed. This includes collocation extraction, word sense disambiguation and concept extraction.

Java WordNet Library

The Java WordNet Library (JWNL)[5] is a Java API for accessing the WordNet dictionary. JWNL makes it for example possible to look up words in WordNet, get their different senses, and find related words through one of a number of relations. For more information on JWNL, please see the JWNL project homepage.

JWNL requires that the dictionary files from WordNet 2.0 are present, these can be found on the WordNet homepage[6]. In addition, they are included along with the source code of this project.

JWNL is used by all the new text preprocessing operations. It is also used in the new rule rating algorithm. How it is used will be described in more detail in the individual operations.

Collocation Extraction

This operation was created to make it possible to identify and extract collocations, presented in Subsection 3.5.2, from the documents. In addition, this operation includes the functionality of the old part of speech tagging operation. That is, extract words from the documents based

[5] http://sourceforge.net/projects/jwordnet

[6] http://wordnet.princeton.edu/

on their part of speech tag. An overview the types of terms which can be extracted is shown below.

- **Verbs** - verbs can be extracted. Terms tagged as verbs are looked up in WordNet using JWNL. This has two benefits. First, the base form of the verb is returned. This means that verbs like ”walking” and ”walked” are both extracted as ”walk”, and the feature dimensionality of the text is reduced. Second, terms which are tagged as verbs but are not present in WordNet, most likely due to erroneous tagging by the tagger, can be removed.
- **Adjectives** - adjectives can also be extracted. As with verbs, adjectives are looked up in WordNet using JWNL to find the base form, and to remove terms which are most likely tagged wrong.
- **Numbers** - Terms tagged as numbers are extracted as is.
- **Adverbs** - Adverbs are also extracted as they are.
- **Proper Nouns** - Proper nouns are extracted as either single terms, or if they appear together with other proper nouns, as a compound of two or more proper nouns or prepositions. Examples include ”Wall Street”, ”President George Bush” or ”Ministry of Defence”. These are not looked up in WordNet.

When it comes to words tagged as noun, these are extracted in one of two ways. First, if a noun appears by itself in a text, it is looked up in WordNet, both to see if it is present and to find its base form. Second, if a noun appears together with another noun, with no sentence dividers between them, they are considered a collocation. Their score are calculated using the chi-square method, presented in Subsection 3.5.2. All collocation candidates with a score over a set critical value, is extracted as collocations. The operation uses a probability level of 0.05. With one degree of freedom, this gives a critical value of 3.841.

The collocations resulting from this operation are generally good. Consider for example the collocations ”defence secretary”, ”car bomb”, ”insurance company”, ”cruise missile” and ”software giant”. There are however also some collocations which do not make much sense, for example ”wont share”, ”suicide car”, ”eleventh female” and ”shadow defence”. This can be caused by at least a couple of reasons. First, a word may be tagged erroneously, but the word is present in WordNet as a noun, and is therefore used as a noun, for example the word ”wont”. Second, the collocation extracted is part of a larger collocation, and therefore does not make much sense by itself. ”Suicide car” is for example part of the larger phrase ”suicide car bomber”.

The examples shown above are collocations extracted from the original Financial Times dataset containing 37 documents. Which dataset is being used will have an impact on the collocations extracted. In addition, the domain of the dataset will influence the results. Many of the collocations extracted from Financial Times can be said to relate to the financial or business domain.

Word Sense Disambiguation

A WSD operation has been implemented in the TTM Testbench. This operation implements both the Lesk algorithm and the adapted Lesk algorithm, which were presented in Section 3.4. Which algorithm to use, is decided by the user. The operation utilizes JWNL for accessing WordNet to retrieve the glosses of the word being disambiguated and its context words.

Only single nouns are disambiguated in this operation. The user is able to specify how many of the words in the context should be used in the disambiguation process. The context words consists of other nouns, and verbs and adjectives if these were extracted during the collocation operation.

To create the extended gloss in the adapted Lesk algorithm, the algorithm uses the glosses of related words in WordNet, these relations were presented in 3.4. To create the extended gloss for a noun, the following related words are used.

- **Hypernyms**
- **Hyponyms**
- **Meronyms**
- **Holonyms**

As mentioned earlier, verbs may also be used for disambiguating a word. The related words used when creating the extended gloss for a verb are shown below.

- **Hypernyms**
- **Troponyms**

When adjectives are used, the extended gloss is generated by adding the gloss of adjectives linked by the similar-to relation. In addition, collocations which are present in WordNet, for example ”interest rate”, are also used in the disambiguation process, and these words are treated as nouns.

To determine the sense of a word, the gloss, or extended gloss, of each of the senses of the word is compared with the gloss, or extended gloss, of all the senses of the words in the surrounding context. The sense with the highest number of matches is returned as the most likely sense for the word being disambiguated.

In addition, the user is able to specify whether verbs and adjectives should be kept. This is because verbs and adjectives can be useful as context words in the disambiguation process, but may not be wanted in the rule mining process.

Concept Extraction

This operation was implemented to extract concept-level document features from the documents. This is done by using JWNL to utilize the hierarchical structure of WordNet. The concept extraction operation is dependent on WSD, since a word may have different senses, and these are linked to different synsets. The operation has three methods for finding concepts in a document. These are described in the following.

First, WordNet contains a relation called *category*. This relation links a synset to a higher-level category, where the category is represented by another synset. An example of this is that ”basic training” is linked to the category ”military”. By exploring this relation for each disambiguated word, it is possible to extract a set of categories which are descriptive of the contents of a document. Note however that only a limited set of the synsets in WordNet are linked to a category.

The second method of finding concepts in a document does this by finding common parent synsets of the words in the document. This is done for each combination of disambiguated nouns in the texts. If the distance between the two words is below or equal to a user-specified threshold, the common parent synset is extracted as a concept. As an example of this, see Figure 4.3, which shows a part of the WordNet hierarchy. As the figure shows, ”yen” and ”euro” has ”monetary unit” as a common parent. Depending on the distance threshold, this may be extracted as a concept.

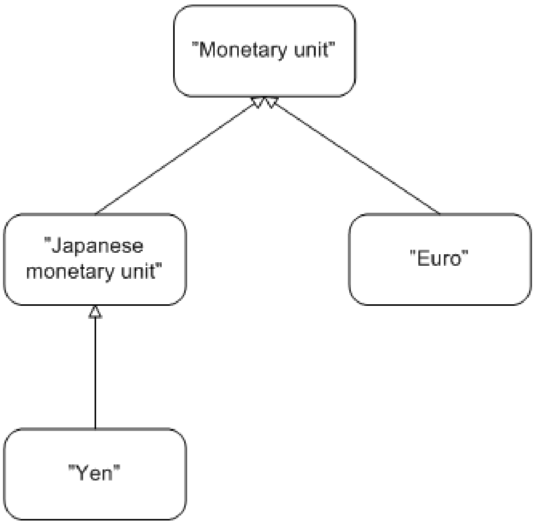


Figure 4.3: Subset of WordNet hierarchy

Finally, if no concepts was found using the two methods presented above, the user can specify that the parent node(s) of a word is to be extracted in addition to the word. This is found using the hypernym-relation. Recall Figure 4.3, if only ”euro” is present in document, ”monetary unit” can be extracted. This method may however result in very high feature dimensionality, and increase the complexity in the rule mining process.

In addition, this operation tries to resolve the problem of synonyms in the text. This is done by replacing disambiguated words with the two first words in the synset it belongs to. The reason for using two words instead of only one, is that this may lead to more meaningful terms. For example, if the word ”auto” is present in a document, and it belongs to the following synset {car, auto, automobile, machine, motorcar}, ”auto” is replaced with the term ”car/auto”. All words in the document collection which belong to this synset will therefore be represented by this term.

4.2.3 Rating of Association Rules

The new rule rating method is based on the semantic similarity measure JCn, presented in Subsection 3.6.1. The measure has not been implemented from scratch, but the Java WordNet Similarity (JWNS) library [7] has been used. In addition to the JCn measure, the library is also able to calculate the Lin measure [25]. For this project however, only the JCn measure will be used. Similar to JWNL, JWNS needs access to the WordNet 2.0 dictionary files.

This measure is calculated after the association rules have been mined. The score of an association rule is calculated as the average semantic similarity between the words in the antecedent

[7] ttp://nlp.shef.ac.uk/result/software.html

and the consequent of the rule. However, note that it is only possible to calculate semantic similarity with disambiguated nouns or collocations which are present in WordNet. This is because the similarity is calculated between synsets, and the sense is needed to know which synset a word is present in.

The semantic similarity can then be used to rank the association rules. The higher the score, the more semantic similar the words in the antecedent and the consequent of the rule are. The rules with the lowest scores can therefore be considered interesting. Whether this actually is the case or not, will be discussed in the next chapter.

4.3 Summary

This chapter has presented the new additions to the TTM Testbench, both in terms of new document collections, new text preprocessing operations, and a new method for rating the association rules. An overview of the experiments and the results of these are given in the next chapter.

Chapter 5

Experiments and Results

This chapter will present the experiments performed in this project. The objective of the experimentation is to see whether any of the goals listed in Section 1.2 are met. The motivation for these experiments is therefore the following.

- To see if a larger document collection can lead to more interesting association rules.
- To see if collocation extraction, word sense disambiguation and concept extraction leads to meaningful and interesting association rules.
- To see whether a semantic similarity measure like JCn can help find the most interesting association rules.

The chapter starts by giving an overview of the experiment setup, this involves details about the hardware used, the parameters and the document collection. Then, the experiments are presented. Finally, an overview of the results is given.

5.1 Experiment Setup

The experiments will be run on a remote server, using the TTM Testbench without a graphical user interface. The application requires that the J2SE Runtime Environment 5.0 is installed. The remote server has the following specifications.

- **Operating System** - FreeBSD
- **CPU** - Intel Core 2 Duo 6600
- **Memory** - 4 GB

The experiments performed in this project will use all three new methods for extracting document features. Filtering and weighting will not be used, since the IDF part of TF-IDF dampens the weight of terms which appear in many documents. This may not be wanted, since association rules containing frequent terms in some cases can be interesting. All rules will be rated using the semantic similarity measure. However, only rules containing at least one disambiguated word on each side of the rule will get a score. The process is illustrated in Figure 5.1.

Which document collection is being used for the experiments, will be covered in the description of each experiment. Both the new Financial Times and the BBC Business datasets will be used. The two datasets have been reduced somewhat, originally they contained two version per day,

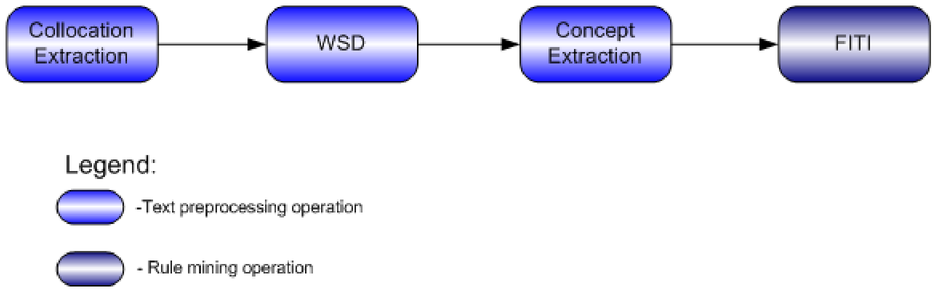


Figure 5.1: The operation process for the experiments

but only one version per day will be used. This means that the new Financial Times dataset contains 107 documents, and the BBC dataset contains 79 documents. Note that the TTM Testbench sees each individual news item as one document, where news items from the same version of the web page gets the same time stamp. Association rules will be found which span across texts with different timestamps.

The same parameters will be used for the three new text operations in all experiments. These are based on initial experiments with the Testbench. The FITI and filter parameters will however vary between the experiments, these are therefore presented in the description of each experiment.

The parameters for the new operations are given below, these operations were described in Subsection 4.2.2.

- **Collocation Extraction**
  - Only verbs and adjectives are extracted in addition to collocations.
- **Word Sense Disambiguation**
  - Context size - 3 words on each side of the word being disambiguated, 6 in total.
  - Adapted Lesk algorithm will be used.
  - Verbs and adjectives are not kept after the disambiguation process.
- **Concept Extraction**
  - Maximum distance in the WordNet hierarchy is 5 (this includes the words themselves).
  - Parent nodes of words with no concepts are not added.
  - Original terms are not kept when a concept is found.

The following terms will therefore be extracted from each document and used in the rule mining process.

- Collocations
- Proper nouns and proper noun groups.
- Common parents between terms in the same document.
- Categories
- Disambiguated nouns with no common parent or category.

- Nouns which have not been disambiguated.

All experiments will use all extracted terms without any filtering, except experiment 1 which is a comparison with the experiment performed last fall. The details for this experiment will be described in Section 5.3.

5.2 Evaluation Criteria

Automatically deciding if a rule is interesting or not, is difficult, if not impossible. In [14], a pattern is defined as interesting if the following conditions are met:

- **Meaningful** - Humans can easily understand the pattern.
- **Valid** - It is valid on new data with some degree of certainty.
- **Useful** - The user is able to act upon it.
- **Novel** - Validates a hypothesis the user sought to confirm.

The main focus in this project will be to see if the association rules and their items are meaningful. In addition, the rules found using the different datasets will be compared to see if there are association rules present in both sets. The two last criteria will not be considered here since these are highly dependent on the background knowledge of the user.

It will also be discussed whether there is any difference between rules with a high semantic similarity and rules with low semantic similarity. Recall that semantic similarity is a measure for calculating the similarity of of a rule based on the distance between the terms in WordNet, and the information content of the terms, semantic similarity was described in Subsection 3.6.1. The idea is that rules with low semantic similarity are more interesting than those with high similarity.

For all experiments, a subset of 15 rules will be presented. These are the 5 first rules with no semantic similarity value, the 5 rules with lowest semantic similarity, and the 5 rules with the highest semantic similarity. For the full results, please see Appendix C. The values for support, confidence and semantic similarity will be left out because of space issues. The rule numbers in the rules presented for the individual experiments refer to their number in the full result set. Keep in mind that it is not possible to calculate the semantic similarity of rules not containing any disambiguated terms, concepts or categories, these will therefore get a semantic similarity of zero and thus appear first in the result set.

The terms present in the rules will sometimes include the symbol #, this is used to indicate the sense number of the term in WordNet. Another symbol which may appear, is "/nnp". This means that the term is a proper noun.

5.3 Experiment 1: Comparison with Fall 2006 Project

This experiment will use the same document collection as the project last fall, this means the Financial Times collection consisting of 37 versions of the front page. This is done to compare the rules found using the new methods with the rules found last fall. The parameters for the FITI algorithm will therefore be the same as those used in that project. These are listed in Table 5.1.

Table 5.1: FITI parameters

Parameter	**Value**
Minimum support	0.1
Maximum support	0.5
Minimum confidence	0.5
Maximum confidence	1.0
Maxspan	3
Max set size	3

In addition, the 4 first terms from each document was selected for rule mining, the results from the 2006 fall project can be seen in Appendix C.1, the result consists of 133 rules.

5.3.1 Results and Discussion

To use the 4 first terms to represent each document, may not be the best method for selecting which terms to use in the rule mining process. This was confirmed by the first experiment, where no rules was found. A reason for this may be that when the N first terms are selected, these terms are not necessarily the most descriptive terms for the document. They may also be present in only a few of the documents, and thus have too low support.

It was therefore decided to run the experiment again, this time with weighting of the terms to select the terms with the highest TF-IDF score. When the process was run with respectively the 4 and 5 highest ranked terms, no rules where found. This may be due to the IDF part of the TF-IDF measure, since this part dampens the weight of terms present in several documents. But terms which only appear in one or few documents, may not have high enough support to be present in an association rule.

The experiment was therefore run with all terms. The rationale for doing this is that only nouns, collocations and concepts are extracted by the feature extraction operations, which are all potentially meaningful. The result of this was 347 rules, shown in Appendix C.2.

As one can see, a large number of association rules is generated. This will make it hard for a user to identify the really interesting rules, if there are any. A small selection of the association rules is shown below, note that support, confidence and similarity is left out because of lack of space.

```
The 5 first rules with no semantic similarity:
Rule 1   - {(’plan/program#1’, 0)} -> {(’iraq/nnp’, 1)}
Rule 2   - {(’commercial_enterprise/business_enterprise#2’ ’company#1’, 0)} -> {(’iraq/nnp’, 1)}
Rule 3   - {(’year#3’ ’europe/nnp’, 0)} -> {(’iraq/nnp’, 1)}
Rule 4   - {(’president_of_the_united_states/united_states_president#1’, 0)} -> {(’iraq/nnp’, 1)}
Rule 5   - {(’consequence/effect#1’ ’economy/economic_system#1’, 0)} -> {(’iraq/nnp’, 1)}

The 5 rules with the lowest semantic similarity:
Rule 128 - {(’yen#2’, 0)} -> {(’law/jurisprudence#2’, 1)}
Rule 129 - {(’depository_financial_institution/bank#1’, 0)} -> {(’dollar#1’, 2)}
Rule 130 - {(’quarter#6’, 0)} -> {(’law/jurisprudence#2’, 1)}
Rule 131 - {(’iraq/nnp’ ’euro#1’, 0)} -> {(’law/jurisprudence#2’, 1)}
Rule 132 - {(’euro#1’, 0)} -> {(’law/jurisprudence#2’, 1)}

The 5 rules with the highest semantics similariy:
```

```
Rule 343 - {('occupation/business#1', 0)} -> {('market/marketplace#1', 1)}
Rule 344 - {('country/state#1' 'government/governing#3', 0)}
 -> {('commercial_enterprise/business_enterprise#2', 1)}
Rule 345 - {('monetary_unit#1', 0)} -> {('yen#2', 1)}
Rule 346 - {('euro#1' 'europe/nnp', 0)} -> {('monetary_unit#1', 1)}
Rule 347 - {('time_period/period_of_time#1', 0) } -> {('year#3', 2)}
```

As the rules above show, many of the terms included in the rules are meaningful, and the user can therefore make sense of the discovered rules. There are however also rules where it is more difficult to immediately understand the meaning of the terms, consider for example rule 343 above. This rule contains the term "market/marketplace", which can mean either a physical location in a city, or the world of commercial activity. For the user to be able to understand what is meant, he will manually have to access WordNet to find the description of the specified sense (denoted by #1).

One aspect that becomes clear when inspecting the rules is that it is easier to understanding the meaning of the items when they are represented by two synonyms. As an example, see rule 129. Here the item "depository_financial_institution/bank" is present. Because a synonym is present, the rule is more meaningful than if for example only "bank" was present.

Whether semantic similarity is able to distinguish between interesting and uninteresting rules or not, is difficult to decide. The reason for this is that it is not entirely clear what an interesting association rule would look like when mining for association rules in web newspapers. But when looking at the rules with the highest and lowest semantic similarity, it can be argued that the rule with the lowest score is probably more interesting since the one with the highest score contains terms related to time in both the antecedent and the consequent.

A thing which can be noted about the results is that some rules contains examples of erroneous part of speech tagging. See for example rules number 46 and 47 (not shown above), these rules include the term "dollar" with the tag "/nnp" indicating that it is a proper noun, but this is not correct. Such rules can be considered noise.

Note that the following four experiments will not use weighting and filtering of the terms.

5.4 Experiment 2: New FT

This experiment will use the same FITI-parameters as experiment 1, these are shown in Table 5.1. The difference from experiment 1, is that this experiment will use the new Financial Times dataset. This is done to see if there is any difference in the association rules discovered, and if these can be said to be more interesting than those found using a smaller dataset.

5.4.1 Results and Discussion

The result of this experiment was 56 rules, these are presented in Appendix C.3. The reason for getting a lot less rules than in experiment 1 may be that a confidence of 0.1 may be too high since it means that for an item to be large it has to appear in 10% of the megatransactions. For this dataset, this means 10% of 105 megatransactions, which is about 10.

Similar to experiment 1, the rules found here contain meaningful terms, and the rules can therefore be said to be meaningful. It is also possible to see that a lot of the same terms are present here, as in experiment 1, for example "military/armed forces#1", "market/marketplace#1" and "iraq/nnp". No equal rules have been found in the two first experiments, but the presence of similar terms in the association rules may give an indication that it is possible. A subset of rules is given below.

```
The 5 first rules with semantic similarity:
Rule 1  - {('europe/nnp', 0)} -> {('market/marketplace#1', 1)}
Rule 2  - {('china/nnp', 0)} -> {('military/armed_forces#1', 1)}
Rule 3  - {('russia/nnp', 0)} -> {('military/armed_forces#1', 1)}
Rule 4  - {('iraq/nnp', 0)} -> {('military/armed_forces#1', 1)}
Rule 5  - {('uk/nnp', 0)} -> {('military/armed_forces#1', 1)}

The 5 rules with the lowest semantic similarity:
Rule 21 - {('china/nnp' 'market/marketplace#1', 0)} -> {('military/armed_forces#1', 1)}
Rule 22 - {('market/marketplace#1', 0)} -> {('military/armed_forces#1', 2)}
Rule 23 - {('market/marketplace#1', 0)} -> {('military/armed_forces#1', 1)}
Rule 24 - {('china/nnp' 'market/marketplace#1', 0)} -> {('military/armed_forces#1', 2)}
Rule 25 - {('company#1', 0)} -> {('market/marketplace#1', 2)}

The 5 rules with the highest semantic similarity:
Rule 52 - {('president_of_the_united_states/united_states_president#1' 'head/chief#4', 0)}
          -> {('investor#1', 2)}
Rule 53 - {('head/chief#4', 0)} -> {('investor#1', 2)}
Rule 54 - {('depository_financial_institution/bank#1', 0)} -> {('military/armed_forces#1', 2)}
Rule 55 - {('company#1', 0)} -> {('military/armed_forces#1', 2)}
Rule 56 - {('company#1', 0)} -> {('military/armed_forces#1', 1)}

Potentially interesting rule:
Rule 33 - {('china/nnp', 0) ('president_of_the_united_states/united_states_president#1', 1)}
          -> {('military/armed_forces#1', 2)}
```

When looking at the rules from this experiment, it becomes apparent that rule number 33, shown above, may be considered interesting. Consider for example that there is an article discussing an event in China at time 0, then the next day a related article appears where the US President is mentioned. Finally, at time 2 an article containing military news which is related to the two previous articles appear. It is however difficult to know whether these cases are related, or just coincidental. But it gives an indication that it may in fact be possible to detect interesting temporal relationships between news items from different versions of the front page of a web newspaper.

As in the first experiment, there is no apparent difference in interestingness between the lowest and highest ranked association rules, respectively rule 21 and rule 56, based on the semantic similarity. A reason for this may be that it is difficult to manually decide which rules are interesting.

5.5 Experiment 3: BBC

This experiment will use the same parameters for the FITI algorithm as the two previous experiments, but it will use the BBC Business dataset. One of the goals of this experiment will be to see if there are similar association rules which appear in two distinct datasets. Such rules may be considered interesting, and a base for further investigation. There is a potential for discovering similar rules since both the new Financial Times and the BBC Business datasets were gathered in the same time period, and both may therefore contain news articles describing the same events.

5.5.1 Results and Discussion

This experiment resulted in 21 rules, these are shown in AppendixC.4. The number of rules here is also less than in experiment 1, and the minimum support threshold of 0.1 is likely to have caused this. In this dataset, for an item to be large it has to appear in about 7 megatransactions since the total number of megatransactions is 77.

As with the previous experiments, most of the terms in the association rules are meaningful and helps the user make sense of the rules. One thing to note about the terms is that these vary slightly from the ones in experiments 1 and 2. A reason for this may be that the vocabulary used in the newspapers vary since BBC Business is an English newspaper, and Financial Times is American. Another thing worth noting, is that 14 of the 21 association rules contain the terms ”UK” or ”EU”, which may not surprising considering that BBC Business is European, and therefore may contain mostly news regarding English or European events. A small subset of the discovered association rules is shown below, here all the rules with a semantic similarity value is shown since there are only eight of these.

```
The 5 first rules with no semantic similarity:
Rule 1  - {('net_income/net#1', 0)} -> {('uk/nnp', 2)}
Rule 2  - {('eu/nnp', 0)} -> {('uk/nnp', 1)}
Rule 3  - {('net_income/net#1', 0)} -> {('uk/nnp', 1)}
Rule 4  - {('rate#2', 0)} -> {('uk/nnp', 2)}
Rule 5  - {('rate#2', 0)} -> {('eu/nnp', 1)}

The 8 rules with semantic similarity:
Rule 14 - {('drop/dip#3', 0)} -> {('firm/house#1', 1)}
Rule 15 - {('uk/nnp' 'net_income/net#1', 0)} -> {('sale/cut-rate_sale#4', 1)}
Rule 16 - {('uk/nnp', 0) ('net_income/net#1', 1)} -> {('sale/cut-rate_sale#4', 2)}
Rule 17 - {('rate#2', 0)} -> {('net_income/net#1', 1)}
Rule 18 - {('uk/nnp', 1) ('net_income/net#1', 0)} -> {('firm/house#1', 2)}
Rule 19 - {('firm/house#1', 1) ('uk/nnp', 0)} -> {('net_income/net#1', 2)}
Rule 20 - {('occupation/business#1', 0)} -> {('net_income/net#1', 1)}
Rule 21 - {('depository_financial_institution/bank#1', 0)} -> {('firm/house#1', 1)}
```

It is still not possible to determine whether semantic similarity is able to distinguish between interesting and uninteresting association rules. Even though both the terms in the association rule with the highest score (rule 21) relate to finance/business, and therefore may be known in advance. The problem is as previously, to be able to clearly specify manually that a rule is interesting.

5.6 Experiment 4: New FT With Adjusted FITI Parameters

This experiment will use the new Financial Times dataset. A problem with the previous experiment on this dataset was that potentially interesting association rules may have been left out because of the relatively high minimum support threshold. This experiment will therefore use a slightly lower threshold. The threshold is set at 0.05, which means that large items need to appear in only about 5 megatransactions. In addition, the minimum confidence threshold has been raised. The rationale behind this is that for a rule to be interesting is has to be strong, and rules with low confidence might only be present by chance. These new parameters for the FITI algorithm are shown in Table 5.2.

Table 5.2: FITI parameters

Parameter	Value

Minimum support	0.05 (0.065)
Maximum support	0.5
Minimum confidence	0.7
Maximum confidence	1.0
Maxspan	3
Max set size	3

5.6.1 Results and Discussion

It soon became apparent that using such low support resulted in a huge increase in the run time of the rule mining algorithm. After executing for over 24 hours, the experiment was aborted, and the minimum support threshold was raised slightly. The new threshold was set to 0.065.

The result of the experiment was 71 rules, the full result set is shown in Appendix C.5. Similar to previous experiments, terms present in the association rules are mostly meaningful, but it is still not obvious which, if any, of the association rules are interesting. A subset of the rules is shown below.

```
The 5 first rules with no semantic similarity value:
Rule 1  - {('america/nnp', 0)} -> {('market/marketplace#1', 1)}
Rule 2  - {('ti/nnp', 0)} -> {('military/armed_forces#1', 2)}
Rule 3  - {('uk/nnp' 'iraq/nnp', 0)} -> {('military/armed_forces#1', 2)}
Rule 4  - {('president/chairman#4', 0)} -> {('uk/nnp', 2)}
Rule 5  - {('europe/nnp' 'uk/nnp', 0)} -> {('military/armed_forces#1', 1)}

The 5 rules with the lowest semantic similarity:
Rule 15 - {('hedge_fund/hedgefund#1', 0)} -> {('military/armed_forces#1', 2)}
Rule 16 - {('stock_exchange/stock_market#1', 0)} -> {('military/armed_forces#1', 2)}
Rule 17 - {('crisis#1', 0)} -> {('military/armed_forces#1', 1)}
Rule 18 - {('election#1', 0)} -> {('investor#1', 2)}
Rule 19 - {('china/nnp' 'election#1', 0)} -> {('investor#1', 2)}

The 5 rules with the highest semantic similarity:
Rule 67 -  {('company#1' 'president_of_the_united_states/united_states_president#1', 0)}
 -> {('military/armed_forces#1', 1)}
Rule 68 -  {('president_of_the_united_states/united_states_president#1' 'head/chief#4', 0)}
     -> {('investor#1', 2)}
Rule 69 -  {('uk/nnp' 'head/chief#4', 0)} -> {('investor#1', 2)}
Rule 70 -  {('uk/nnp', 1) ('head/chief#4', 0)} -> {('investor#1', 2)}
Rule 71 -  {('group/grouping#1', 0)} -> {('military/armed_forces#1', 2)}
```

In this experiment, there is also a potential interesting rule. Consider rule 17, if the news item which contains ”crisis” and the news item which contains ”military/armed forces” are related to the same event. For example, that a crisis of some sort happens and is discussed in an article, and the next day an article discusses some sort of military response to this event. As mentioned in experiment 2, it is however difficult to conclude that this is the case.

Some of the rules found in this experiment, was also found in experiment 2. These are the rules from experiment 2 which had a confidence value above 0.70. But new rules are added to the result set which have lower support than those in experiment 2.

When comparing the rule with the highest semantic similarity value to the one still no conclusions can be made regarding the interestingness of the respective rules.

5.7 Experiment 5: BBC With Adjusted FITI Parameters

This experiment will use the BBC Business dataset with modified FITI parameters. The parameters are shown in Table 5.3.

Table 5.3: FITI parameters

Parameter	Value
Minimum support	0.06
Maximum support	0.5
Minimum confidence	0.7
Maximum confidence	1.0
Maxspan	3
Max set size	3

5.7.1 Results and Discussion

The result of this experiment was 43 rules, shown in Appendix C.6. As with all the previous experiments, most of the items present in the association rules make sense and are possible to understand. The problem is that it is still not clear whether the rules are interesting, or if some rules are more interesting than others. An overview of some of the rules is given below.

```
The 5 first rules with no semantic similarity value:
Rule 1  - {(’rate#2’ ’uk/nnp’, 0)} -> {(’eu/nnp’, 1)}
Rule 2  - {(’rate#2’ ’sale/cut-rate_sale#4’, 0)} -> {(’eu/nnp’ , 1)}
Rule 3  - {(’india/nnp’, 1) (’uk/nnp’, 0)} -> {(’firm/house#1’, 2)}
Rule 4  - {(’uk/nnp’ ’robert_peston/nnp’, 0)} -> {(’net_income/net#1’, 1)}
Rule 5  - {(’barclays/nnp’, 0)} -> {(’net_income/net#1’, 1)}

The 5 rules with the lowest semantic similarity:
Rule 25 - {(’takeover_bid#1’, 0)} -> {(’net_income/net#1’, 2)}
Rule 26 - {(’uk/nnp’, 0) (’stock_exchange/stock_market#1’, 1)} -> {(’firm/house#1’, 2)}
Rule 27 - {(’sale/cut-rate_sale#4’ ’barclays/nnp’, 0)} -> {(’net_income/net#1’, 1)}
Rule 28 - {(’sale/cut-rate_sale#4’ ’barclays/nnp’, 0)} -> {(’net_income/net#1’, 2)}
Rule 29 - {(’sale/cut-rate_sale#4’, 1) (’china/nnp’, 0)} -> {(’firm/house#1’, 2)}

The 5 rules with the highest semantic similarity:
Rule 39 - {(’depository_financial_institution/bank#1’, 1) (’uk/nnp’, 0)}
          -> {(’firm/house#1’, 2)}
Rule 40 - {(’depository_financial_institution/bank#1’, 0)} -> {(’firm/house#1’, 1)}
Rule 41 - {(’depository_financial_institution/bank#1’ ’china/nnp’, 0)}
          -> {(’firm/house#1’, 1)}
Rule 42 - {(’depository_financial_institution/bank#1’ ’uk/nnp’, 0)}
          -> {(’firm/house#1’, 1)}
Rule 43 - {(’depository_financial_institution/bank#1’ ’japan/nnp’, 0)}
          -> {(’firm/house#1’, 1)}
```

One thing which is interesting to note, is that only one of the rules found in this experiment was also present in experiment 3, rule 40 in this experiment and rule 21 in experiment 3. This shows that the support and confidence thresholds used in the mining process has a huge effect on the resulting rules, and future work on mining temporal association rules should be be aware of this.

There is still no clear difference in interestingness between rules with low semantic similarity and high semantic similarity. If one looks at these rules, rules 25 and 43 above, it is clear that both contain business and finance terminology, but it is not possible to say anything about their interestingness.

5.8 Summary

As the experiments show, the main problem of mining textual association rules from web newspapers is that it is difficult, if not impossible, to clearly see which rules are interesting. However, the rules found in this project using the new document feature extraction operations can be said to make sense. A reason for this is that synonyms are added to words if available, and thus ”head/chief” is easier to understand than only the word ”head”.

When it comes to using semantic similarity for rating association rules, it is still an open question whether this can lead to good results. The reason for this is that identifying interesting rules is difficult, and it is therefore not possible to say if rules with low semantic similarity are more interesting than rules with high similarity.

A problem which has not been discussed earlier, is the problem of assigning the correct sense to a word. In [3], the precision of the adapted Lesk algorithm was reported to be about 35%, a similar precision was found in this project when 8 random texts from the original Financial Times collection was manually inspected after the disambiguation had been performed. The results of this informal inspection is shown in Table 5.4.

Table 5.4: Results from informal evaluation of the WSD process

Description	**Value**
Nouns in total	34
Correct sense	13
Incorrect sense	14
Undecided	7

As the table shows the precision is close to 35%. In addition, in 7 cases it was not possible to determine if the correct sense was assigned to a word. The reason for this is that the senses in WordNet are very fine-grained, and it is difficult to spot the difference. The problem of too fine grained senses in WordNet is also reported by [15].

The implication of this problem to the results of this project, is that care must be taken when looking at the association rules since some of the terms may be present due to erroneous word sense disambiguation. However, when looking at the results, one can see that many of the terms are related to the financial and business domain of Financial Times and BBC Business. In addition, words which are disambiguated incorrectly may be filtered out during the rule mining process because their support in the document collection as a whole is too low.

One of the problems with interestingness when mining for association rules in web newspapers, is that what may seem like an interesting rule, really is a coincidence. Consider for example the rule given in the problem description, namely $\{(\text{'Bomb'}, 0)\} \rightarrow \{(\text{'Terror'}, 1)\}$. At first glance, this rule may seem interesting. But after further inspection it may become clear that the news article containing the word 'terror' is in no way related to the article containing 'bomb', instead it may relate to a totally different event and the association rule is totally coincidental.

Another problem with textual association rule mining is that the really interesting patterns may be present only once in a collection, but using such a low minimum support threshold will lead to a lot of coincidental rules being returned, in addition the run time of the rule mining algorithm will increase. It is therefore vital that the rules are filtered out before they are presented to the user, so that only the potentially interesting ones are returned. Whether the semantic similarity measure is able to do this, is still an open question.

Chapter 6

Further Work

There are several possible future directions for work on the TTM Testbench. One of the most promising may be to use the application in a domain where the interestingness of association rules is more clear. An example of such a domain can be medicine.

As mentioned earlier, the relatively low precision of the adapted Lesk algorithm can lead to credibility issues for the association rules since terms may be present because of erroneous results in the disambiguation process. This could be solved by either studying new algorithms for WSD, or to use another hierarchical knowledge structure where the senses are not so fine-grained. It is for example possible to use a domain specific hierarchy where only one sense or meaning exists for each word, so that every instance of a word in a text is assumed to take the meaning specified in the hierarchy.

It will also be possible to extend the collocation extraction method. Now, only collocations consisting of two nouns are detected, but it may be useful to include some of the other patterns proposed in [19]. Another possibility is to look at entirely different methods and algorithms for extracting document features.

One possibility, which has not been considered by this project, is to look at optimization issues for the various operations implemented in the application. This may particularly be the case for the WSD operation and the FITI algorithm which are both time consuming (depending on the parameters). One method of decreasing the processing time may be to introduce some sort of parallelism in the process, for example through the use of distributed processing.

Chapter 7

Conclusions

Text mining is a field which has emerged due to the huge amounts of textual data available, and the rate of growth. These huge amounts of data make it impossible for humans to manually extract useful knowledge. Tools for performing this task automatically is therefore of great interest.

This project has focused on a special field of text mining, called temporal text mining. The goal has been to discover interesting association rules between news items appearing in different versions of the front page of a web newspaper. The work in this project has continued the work of two previous projects, and has tried to improve the results by further enriching the semantic value of the terms extracted from the different texts. In addition, work has been done to find automatic methods for ranking association rules based on their potential interestingness.

A major part of the work on this master thesis has been to identify potential methods and algorithms related to the identification and extraction of meaningful document terms. WordNet[1] was discovered as a potentially useful source for knowledge about terms appearing in textual documents. A method for extracting simple collocations consisting of two nouns was implemented. Word sense disambiguation was also implemented to identify the meaning of the words, this was used to find relationships between terms in the hierarchy of WordNet so that more general concepts could be extracted. In addition, terms extracted from the news items were represented by two synonyms to increase the meaningfulness of the terms present in the association rules.

Some work was done to gain knowledge about different automatic methods and measures for ranking association rules. A measure which calculates semantic similarity between the terms in the association rules was identified as a promising approach. This was incorporated in the TTM Testbench application by using an external library which uses the WordNet hierarchy as part of the calculation.

There was performed 5 experiments in this project. These were carried out both to compare the meaningfulness of the association rules found using the new methods to those found in the previous projects, and to see what effect larger document collections had on the results. The results show that particularly adding synonyms to the terms makes the rules more understandable. However, the problem is to identify which rules are really interesting, and this became a problem when evaluating whether or not semantic similarity is capable of ranking the rules

[1] http://wordnet.princeton.edu/

based on their interestingness since there was no way of telling if the rules with low semantic similarity were more interesting than the ones with high semantic similarity.

As a final remark, it can be said that to perform association rule mining on web newspapers may be of limited value because of the difficulty in determining the interestingness of a rule. A more suited application may be to perform the rule mining on documents from a more limited domain. Medicine is an example of such a domain, where association rules modeling relationships between diseases or treatments could be found. The results could then be presented to medical personnel to get feedback on which rules are interesting.

Appendix A

The TTM Testbench

This appendix shows the package diagram of the TTM Testbench, and a short description of each package is given. The packages are shown in Figure A.1.

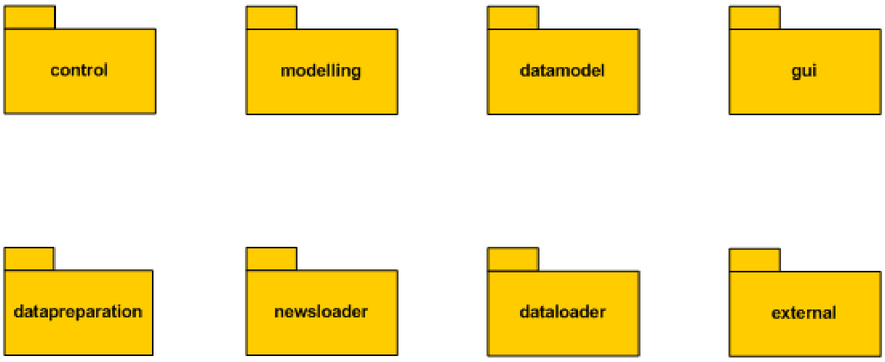


Figure A.1: Package diagram for the TTM Testbench

A short description of the packages is given below.

- **control** - This package handles the control of the application.
- **modelling** - This package contains the classes used by the FITI algorithm. New rule mining algorithms should be put in this package.
- **datamodel** - This package contains the classes used to represent the data model used by the application.
- **gui** - This package contains the graphical user interface of the application.
- **datapreparation** - This package contains all the text preprocessing operations. New operations for extracting document features should be put in this package.
- **newsloader** - This package contains the news item extractor created in the project last fall.
- **dataloader** - This package contains the original method used for extracting news items.
- **external** - This package contains external classes used by the application.

Appendix B

Adding operations to the TTM Testbench

This appendix will give an overview of how new operations can be added to the TTM Testbench, both text preprocessing operations and new rule mining algorithms.

B.1 Adding Text Operations

The possibility to add, or remove, text operations is important for future use of the TTM Testbench. For example, new feature extraction operations.

Text operations are located in the **datapreparation**-package. In addition, a super-class called *TextOperation* is used which all text preprocessing operations need to extend. Finally, a list of possible operations is present in the OperationConfig-class in the control-package.

The process of adding a new text operation is straightforward and the steps are listed below.

- Put your operation in the **datapreparation**-package.
- Let your operation extend the *TextOperation* ckass.
- Add your operation to the list of possible operations in the *OperationConfig*-class in the **control**-package.
- The previous point will need you to make changes in the addConfigOperations-method in the *OperationConfig*-class. This method is used when running the Testbench without GUI.

This was just a brief overview, and inspection of the source code will probably help make the steps presented above more clear.

B.2 Adding Rule Mining Algorithms

In this section, a brief overview is given of how new rule mining algorithms may be included in the application. This could be useful if new rule mining algorithms were to surface in the future.

Rule mining algorithms are located in the **modelling**-package. In addition, rule mining algorithms need to extend a super-class called *ModelOperation*.

The process of adding a new rule mining algorithm consists of the following steps.

- Add the rule mining algorithm to the **modelling**-package.
- The algorithm needs to extend the *ModelOperation* class.
- Make sure the algorithm is able to handle the datamodel used by the application, this is located in the **datamodel**-package. Otherwise the datamodel needs to be modified.
- The algorithm must be added to the list of possible operations in *OperationConfig*-class in the **control**-package.
- If the application is to be used without the GUI, changes must be made to the addConfigOperations-method in the *OperationConfig*-class.

As with adding text preprocessing operations, the steps will probably become clearer after inspecting the source code.

Appendix C

Results

This appendix contains the complete set of rules found in the experiments performed in this project. In addition, it also contains the rules found in one of the experiments last fall.

Association rules are presented on this form: {('Term1' 0), ('Term2', 1)} → {('Term3', 2)}. This means 'term' at time 0 and 'term2' at time 1, indicates 'term3' at time time 2. It may sometimes be the case that the first term has time 1, and the second term time 0, but the meaning of the rule is the same. In addition, the support, confidence and semantic similarity values of the rules are shown.

The symbol # appended to a word indicate the sense number of the word in WordNet(2.0), 'reform#1' for example, means that sense number 1 of "reform" is used. In addition, nouns which are not disambiguated are labelled "/nn", and proper nouns are marked with "/nnp".

An important thing to note about the association rules, is that sometimes a term is represented by two words, separated by "/". This means that the words on either side of the divider are synonyms. For example "conflict/struggle".

The rules, except the ones from 2006, are sorted ascending based on their semantic similarity score.

C.1 Results Fall Project 2006

This section shows the results of one of the experiments performed the fall of 2006. Note that these rules also include the cluster number each term in the rule appears in, this is 0 for every term, since the experiment was performed without the use of clustering.

```
********** TTM Testbench Results ***********************************************

Dataset:
Name: FTNews
Time Granularity: days
Language: English
DataSet size: 842

FITI Settings:
Minimum support: 0.1
Maximum support: 0.5
Minimum confidence: 0.5
```

```
Maximum confidence: 1.0
Maxspan: 3
Max set size: 3
Use only K first documents: -1
Prune rules by order of rules: true


Rules: (133)
Rule 1:   {('chief' , 1) 0} -> {('iran' , 2) 0} , support = 0.14 , confidence =
    0.50
Rule 2:   {('world' , 0) 0} -> {('uk' 'bush' , 2) 0} , support = 0.11 ,
    confidence = 0.57
Rule 3:   {('california' , 0) 0} -> {('crisis' , 1) 0} , support = 0.11 ,
    confidence = 0.57
Rule 4:   {('california' , 0) 0} -> {('business' , 2) 0} , support = 0.11 ,
    confidence = 0.57
Rule 5:   {('california' , 0) 0} -> {('business' , 1) 0} , support = 0.14 ,
    confidence = 0.71
Rule 6:   {('rate' , 0) 0} -> {('uk' , 1) 0} , support = 0.11 , confidence =
    0.67
Rule 7:   {('microsoft' , 0) 0} -> {('uk' , 1) 0} , support = 0.11 , confidence
    = 0.50
Rule 8:   {('trade' , 0) 0} -> {('expectations' , 1) 0} , support = 0.11 ,
    confidence = 0.50
Rule 9:   {('wolfgang munchau:' , 0) 0} -> {('profits' , 2) 0} , support = 0.11 ,
      confidence = 0.80
Rule 10:   {('trade' , 1) 0} -> {('eu' , 2) 0} , support = 0.14 , confidence =
    0.63
Rule 11:   {('dollar' , 1) 0} -> {('eu' , 2) 0} , support = 0.14 , confidence =
    0.50
Rule 12:   {('expectations' , 0) 0} -> {('profits' , 1) 0} , support = 0.11 ,
    confidence = 0.57
Rule 13:   {('uk' 'sales' , 1) 0} -> {('bid' , 2) 0} , support = 0.11 ,
    confidence = 0.57
Rule 14:   {('yukos' , 1) 0} -> {('banks' , 2) 0} , support = 0.17 , confidence
    = 0.55
Rule 15:   {('economy' , 1) 0} -> {('minister' , 2) 0} , support = 0.11 ,
    confidence = 0.50
Rule 16:   {('crisis' , 0) 0} -> {('business' , 2) 0} , support = 0.14 ,
    confidence = 0.56
Rule 17:   {('banks' 'yukos' , 1) 0} -> {('business' , 2) 0} , support = 0.11 ,
    confidence = 0.80
Rule 18:   {('banks' , 0) 0} -> {('business' , 2) 0} , support = 0.11 ,
    confidence = 0.50
Rule 19:   {('banks' , 1) 0} -> {('business' , 2) 0} , support = 0.14 ,
    confidence = 0.63
Rule 20:   {('banks' , 1) 0('yukos' , 0) 0} -> {('business' , 2) 0} , support =
    0.14 , confidence = 0.83
Rule 21:   {('yukos' , 0) 0} -> {('business' , 2) 0} , support = 0.19 ,
    confidence = 0.64
Rule 22:   {('crisis' 'yukos' , 1) 0} -> {('business' , 2) 0} , support = 0.11 ,
     confidence = 0.80
Rule 23:   {('crisis' 'yukos' , 0) 0} -> {('business' , 2) 0} , support = 0.11 ,
     confidence = 0.80
Rule 24:   {('eu' 'banks' , 1) 0} -> {('business' , 2) 0} , support = 0.11 ,
    confidence = 0.67
Rule 25:   {('yukos' , 1) 0} -> {('business' , 2) 0} , support = 0.17 ,
    confidence = 0.55
Rule 26:   {('chief' , 0) 0} -> {('microsoft' , 2) 0} , support = 0.17 ,
    confidence = 0.67
```

```
Rule 27:   {('chief' , 0) 0} -> {('economy' , 2) 0} ,  support = 0.14 ,
    confidence = 0.56
Rule 28:   {('trade' , 1) 0} -> {('expectations' , 2) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 29:   {('business' 'banks' , 1) 0} -> {('yukos' , 2) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 30:   {('business' 'banks' , 0) 0} -> {('yukos' , 2) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 31:   {('eu' 'banks' , 0) 0} -> {('yukos' , 2) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 32:   {('forecasts' , 0) 0} -> {('eu' , 1) 0} ,  support = 0.11 ,  confidence
     = 0.67
Rule 33:   {('forecasts' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 34:   {('forecasts' , 0) 0} -> {('sales' , 1) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 35:   {('attack' , 0) 0} -> {('news' , 1) 0} ,  support = 0.14 ,  confidence
    = 0.56
Rule 36:   {('economy' , 1) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,  confidence =
     0.50
Rule 37:   {('bush' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.19 ,  confidence =
    0.50
Rule 38:   {('upbeat' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,  confidence =
    0.80
Rule 39:   {('world' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,  confidence =
    0.57
Rule 40:   {('microsoft' , 1) 0} -> {('uk' , 2) 0} ,  support = 0.14 ,  confidence
     = 0.56
Rule 41:   {('earnings' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.14 ,  confidence
    = 0.71
Rule 42:   {('rate' , 1) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,  confidence =
    0.67
Rule 43:   {('trade' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.14 ,  confidence =
    0.63
Rule 44:   {('sales' 'profits' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 45:   {('bush' 'profits' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 46:   {('profit' , 0) 0} -> {('uk' , 2) 0} ,  support = 0.11 ,  confidence =
    0.57
Rule 47:   {('trade' , 0) 0} -> {('investors' , 2) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 48:   {('ahold' , 1) 0} -> {('investors' , 2) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 49:   {('bank' , 0) 0} -> {('business' , 2) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 50:   {('bank' , 0) 0} -> {('business' , 1) 0} ,  support = 0.14 ,
    confidence = 0.71
Rule 51:   {('expectations' , 1) 0} -> {('profits' , 2) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 52:   {('talks' , 0) 0} -> {('yukos' , 2) 0} ,  support = 0.11 ,  confidence
    = 0.57
Rule 53:   {('ahold' , 0) 0} -> {('investors' , 1) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 54:   {('banks' 'yukos' , 0) 0} -> {('business' , 1) 0} ,  support = 0.11 ,
    confidence = 0.80
Rule 55:   {('banks' , 0) 0} -> {('business' , 1) 0} ,  support = 0.14 ,
    confidence = 0.63
Rule 56:   {('yukos' , 0) 0} -> {('business' , 1) 0} ,  support = 0.17 ,
    confidence = 0.55
```

```
Rule 57:   {('crisis' 'yukos' , 0) 0} -> {('business' , 1) 0} ,  support = 0.11 ,
     confidence = 0.80
Rule 58:   {('eu' 'banks' , 0) 0} -> {('business' , 1) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 59:   {('banks' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,  confidence =
     0.50
Rule 60:   {('policy' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.14 ,  confidence
   = 0.71
Rule 61:   {('economy' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,  confidence
    = 0.57
Rule 62:   {('markets' 'profits' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 63:   {('deal' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.17 ,  confidence =
    0.60
Rule 64:   {('trade' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,  confidence =
     0.50
Rule 65:   {('profits' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.22 ,  confidence
     = 0.62
Rule 66:   {('markets' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.14 ,  confidence
     = 0.63
Rule 67:   {('expectations' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 68:   {('microsoft' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.14 ,
    confidence = 0.63
Rule 69:   {('quarter' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.11 ,  confidence
     = 0.80
Rule 70:   {('attack' , 1) 0} -> {('news' , 2) 0} ,  support = 0.14 ,  confidence
   = 0.56
Rule 71:   {('banks' , 0) 0} -> {('yukos' , 2) 0} ,  support = 0.14 ,  confidence
   = 0.63
Rule 72:   {('banks' , 0) 0} -> {('profits' , 1) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 73:   {('banks' , 0) 0} -> {('yukos' , 1) 0} ,  support = 0.11 ,  confidence
   = 0.50
Rule 74:   {('eu' , 0) 0} -> {('bush' , 1) 0} ,  support = 0.17 ,  confidence =
    0.50
Rule 75:   {('eu' , 0) 0} -> {('business' , 1) 0} ,  support = 0.19 ,  confidence
   = 0.58
Rule 76:   {('business' 'banks' , 0) 0} -> {('yukos' , 1) 0} ,  support = 0.11 ,
    confidence = 0.67
Rule 77:   {('forecasts' , 1) 0} -> {('bush' , 2) 0} ,  support = 0.11 ,
    confidence = 0.80
Rule 78:   {('plan' , 0) 0} -> {('sales' , 1) 0} ,  support = 0.11 ,  confidence =
     0.50
Rule 79:   {('markets' , 0) 0} -> {('sales' , 1) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 80:   {('bush' 'profits' , 0) 0} -> {('sales' , 1) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 81:   {('expectations' , 0) 0} -> {('sales' , 1) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 82:   {('microsoft' , 0) 0} -> {('sales' , 1) 0} ,  support = 0.11 ,
    confidence = 0.50
Rule 83:   {('market' , 0) 0} -> {('profits' , 2) 0} ,  support = 0.11 ,
    confidence = 0.57
Rule 84:   {('yukos' , 0) 0} -> {('profits' , 2) 0} ,  support = 0.19 ,
    confidence = 0.64
Rule 85:   {('bank' , 1) 0} -> {('business' , 2) 0} ,  support = 0.14 ,
    confidence = 0.71
Rule 86:   {('expectations' , 1) 0} -> {('bush' , 2) 0} ,  support = 0.11 ,
    confidence = 0.57
```

```
Rule 87:  {('economy' , 1) 0} -> {('bush' , 2) 0} , support = 0.14 , confidence
    = 0.63
Rule 88:  {('banks' , 1) 0} -> {('bush' , 2) 0} , support = 0.11 , confidence =
    0.50
Rule 89:  {('markets' 'profits' , 1) 0} -> {('bush' , 2) 0} , support = 0.11 ,
   confidence = 0.67
Rule 90:  {('world' , 0) 0} -> {('bush' , 2) 0} , support = 0.14 , confidence =
    0.71
Rule 91:  {('microsoft' , 1) 0} -> {('bush' , 2) 0} , support = 0.17 ,
   confidence = 0.67
Rule 92:  {('profits' , 1) 0} -> {('bush' , 2) 0} , support = 0.22 , confidence
    = 0.62
Rule 93:  {('trade' , 1) 0} -> {('bush' , 2) 0} , support = 0.11 , confidence =
    0.50
Rule 94:  {('earnings' , 0) 0} -> {('bush' , 2) 0} , support = 0.11 ,
   confidence = 0.57
Rule 95:  {('plan' , 0) 0} -> {('bush' , 2) 0} , support = 0.11 , confidence =
   0.50
Rule 96:  {('sales' 'profits' , 0) 0} -> {('bush' , 2) 0} , support = 0.14 ,
   confidence = 0.71
Rule 97:  {('profits' , 0) 0} -> {('bush' , 2) 0} , support = 0.19 , confidence
    = 0.54
Rule 98:  {('markets' , 0) 0} -> {('bush' , 2) 0} , support = 0.11 , confidence
    = 0.50
Rule 99:  {('microsoft' 'economy' , 1) 0} -> {('bush' , 2) 0} , support = 0.11 ,
     confidence = 0.67
Rule 100:  {('attack' , 0) 0} -> {('bush' , 2) 0} , support = 0.14 , confidence
    = 0.56
Rule 101:  {('attack' , 0) 0('profits' , 1) 0} -> {('bush' , 2) 0} , support =
   0.11 , confidence = 1.00
Rule 102:  {('deal' , 1) 0} -> {('bush' , 2) 0} , support = 0.17 , confidence =
    0.60
Rule 103:  {('bid' , 0) 0} -> {('bush' , 2) 0} , support = 0.17 , confidence =
   0.55
Rule 104:  {('quarter' , 1) 0} -> {('bush' , 2) 0} , support = 0.11 ,
   confidence = 0.80
Rule 105:  {('markets' , 1) 0} -> {('bush' , 2) 0} , support = 0.14 ,
   confidence = 0.63
Rule 106:  {('policy' , 1) 0} -> {('bush' , 2) 0} , support = 0.14 , confidence
    = 0.71
Rule 107:  {('profit' , 0) 0} -> {('bush' , 2) 0} , support = 0.14 , confidence
    = 0.71
Rule 108:  {('countries' , 0) 0} -> {('california' , 1) 0} , support = 0.11 ,
   confidence = 0.80
Rule 109:  {('dollar' 'countries' , 0) 0} -> {('california' , 1) 0} , support =
   0.11 , confidence = 0.80
Rule 110:  {('california' , 1) 0} -> {('business' , 2) 0} , support = 0.14 ,
   confidence = 0.83
Rule 111:  {('california' , 1) 0} -> {('crisis' , 2) 0} , support = 0.11 ,
   confidence = 0.67
Rule 112:  {('bush' 'profits' , 1) 0} -> {('attack' , 2) 0} , support = 0.11 ,
   confidence = 0.50
Rule 113:  {('bush' 'profits' , 0) 0} -> {('attack' , 1) 0} , support = 0.11 ,
   confidence = 0.50
Rule 114:  {('dollar' 'countries' , 0) 0} -> {('eu' , 1) 0} , support = 0.11 ,
   confidence = 0.80
Rule 115:  {('eu' , 1) 0} -> {('business' , 2) 0} , support = 0.19 , confidence
    = 0.64
Rule 116:  {('eu' , 1) 0} -> {('bush' , 2) 0} , support = 0.17 , confidence =
   0.55
```

```
Rule 117:   {('countries' , 0) 0} -> {('eu' , 1) 0} , support = 0.11 ,
    confidence = 0.80
Rule 118:   {('trade' , 0) 0} -> {('eu' , 1) 0} , support = 0.14 , confidence =
    0.63
Rule 119:   {('california' , 0) 0} -> {('eu' , 1) 0} , support = 0.11 ,
    confidence = 0.57
Rule 120:   {('dollar' , 0) 0} -> {('eu' , 1) 0} , support = 0.17 , confidence =
     0.55
Rule 121:   {('uk' 'sales' , 0) 0} -> {('bid' , 1) 0} , support = 0.14 ,
    confidence = 0.63
Rule 122:   {('expectations' , 1) 0} -> {('sales' , 2) 0} , support = 0.11 ,
    confidence = 0.57
Rule 123:   {('yukos' , 0) 0} -> {('sales' , 2) 0} , support = 0.17 , confidence
     = 0.55
Rule 124:   {('trade' , 0) 0} -> {('sales' , 2) 0} , support = 0.11 , confidence
     = 0.50
Rule 125:   {('markets' , 0) 0} -> {('sales' , 2) 0} , support = 0.11 ,
    confidence = 0.50
Rule 126:   {('bush' 'profits' , 1) 0} -> {('sales' , 2) 0} , support = 0.11 ,
    confidence = 0.50
Rule 127:   {('microsoft' , 0) 0} -> {('sales' , 2) 0} , support = 0.11 ,
    confidence = 0.50
Rule 128:   {('bush' , 1) 0('profits' , 0) 0} -> {('sales' , 2) 0} , support =
    0.11 , confidence = 0.50
Rule 129:   {('markets' , 1) 0} -> {('sales' , 2) 0} , support = 0.11 ,
    confidence = 0.50
Rule 130:   {('plan' , 1) 0} -> {('sales' , 2) 0} , support = 0.11 , confidence
    = 0.50
Rule 131:   {('banks' , 1) 0} -> {('yukos' , 2) 0} , support = 0.11 , confidence
     = 0.50
Rule 132:   {('yukos' , 0) 0} -> {('banks' , 1) 0} , support = 0.17 , confidence
     = 0.55
Rule 133:   {('banks' , 1) 0} -> {('profits' , 2) 0} , support = 0.11 ,
    confidence = 0.50
```

C.2 Experiment 1: Comparison with Fall 2006 Project

This section shows the results of running the process on the original Financial Times dataset, with all terms extracted.

```
********** TTM Testbench Results *******************************************

Dataset:
Name: FTNews
Time Granularity: days
Language: English
DataSet size: 418

FITI Settings:
Minimum support: 0.1
Maximum support: 0.5
Minimum confidence: 0.5
Maximum confidence: 1.0
Maxspan: 3
Max set size: 3
Use only K first documents: -1
Prune rules by order of time: true
```

```
Rules: (347)
Rule 1:   {('plan/program#1' , 0) } -> {('iraq/nnp' , 1) } , support = 0.14 ,
    confidence = 0.63 , similarity = 0.0000
Rule 2:   {('commercial_enterprise/business_enterprise#2' 'company#1' , 0) } ->
    {('iraq/nnp' , 1) } , support = 0.11 , confidence = 0.67 , similarity =
    0.0000
Rule 3:   {('year#3' 'europe/nnp' , 0) } -> {('iraq/nnp' , 1) } , support = 0.11
    , confidence = 0.50 , similarity = 0.0000
Rule 4:   {('president_of_the_united_states/united_states_president#1' , 0) } ->
    {('iraq/nnp' , 1) } , support = 0.14 , confidence = 0.56 , similarity =
    0.0000
Rule 5:   {('consequence/effect#1' 'economy/economic_system#1' , 0) } -> {('iraq/
    nnp' , 1) } , support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 6:   {('company#1' , 0) } -> {('iraq/nnp' , 1) } , support = 0.19 ,
    confidence = 0.64 , similarity = 0.0000
Rule 7:   {('europe/nnp' , 0) } -> {('iraq/nnp' , 1) } , support = 0.19 ,
    confidence = 0.54 , similarity = 0.0000
Rule 8:   {('country/state#1' 'government/governing#3' , 0) } -> {('iraq/nnp' , 1)
     } , support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 9:   {('europe/nnp' , 0) } -> {('economy/economic_system#1' , 2) } , support
     = 0.19 , confidence = 0.54 , similarity = 0.0000
Rule 10:   {('law/jurisprudence#2' , 0) ('year#3' , 1) } -> {('eu/nnp' , 2) } ,
    support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 11:   {('law/jurisprudence#2' , 0) ('share/portion#2' , 1) } -> {('eu/nnp' ,
    2) } , support = 0.14 , confidence = 0.83 , similarity = 0.0000
Rule 12:   {('law/jurisprudence#2' , 0) ('economy/economic_system#1' , 1) } -> {('
    eu/nnp' , 2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 13:   {('plan/program#1' , 0) } -> {('europe/nnp' , 1) } , support = 0.11 ,
     confidence = 0.50 , similarity = 0.0000
Rule 14:   {('eu/nnp' 'iraq/nnp' , 0) } -> {('economy/economic_system#1' , 1) } ,
     support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 15:   {('bush/nnp' , 0) } -> {('iraq/nnp' , 1) } , support = 0.17 ,
    confidence = 0.67 , similarity = 0.0000
Rule 16:   {('bush/nnp' , 0) } -> {('europe/nnp' , 1) } , support = 0.14 ,
    confidence = 0.56 , similarity = 0.0000
Rule 17:   {('bush/nnp' , 0) } -> {('commercial_enterprise/business_enterprise#2'
    , 1) } , support = 0.19 , confidence = 0.78 , similarity = 0.0000
Rule 18:   {('bush/nnp' , 0) } -> {('economy/economic_system#1' , 2) } , support
    = 0.14 , confidence = 0.56 , similarity = 0.0000
Rule 19:   {('bush/nnp' , 0) } -> {('iraq/nnp' 'commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.14 , confidence = 0.56 ,
    similarity = 0.0000
Rule 20:   {('bush/nnp' , 0) } -> {('economy/economic_system#1' , 1) } , support
    = 0.14 , confidence = 0.56 , similarity = 0.0000
Rule 21:   {('eu/nnp' 'iraq/nnp' , 0) } -> {('country/state#1' , 2) } , support =
     0.11 , confidence = 0.50 , similarity = 0.0000
Rule 22:   {('yukos/nnp' , 0) } -> {('country/state#1' , 2) } , support = 0.11 ,
     confidence = 0.57 , similarity = 0.0000
Rule 23:   {('commercial_enterprise/business_enterprise#2' 'year#3' , 0) } -> {('
    iraq/nnp' , 2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 24:   {('dollar#1' , 0) } -> {('iraq/nnp' , 2) } , support = 0.17 ,
    confidence = 0.60 , similarity = 0.0000
Rule 25:   {('plan/program#1' , 0) } -> {('iraq/nnp' , 2) } , support = 0.17 ,
    confidence = 0.75 , similarity = 0.0000
Rule 26:   {('commercial_enterprise/business_enterprise#2' , 1) ('country/state#1'
     , 0) } -> {('iraq/nnp' , 2) } , support = 0.11 , confidence = 0.50 ,
    similarity = 0.0000
```

```
Rule 27:   {('commercial_enterprise/business_enterprise#2' , 1) ('company#1' , 0)
    } -> {('iraq/nnp' , 2) } , support = 0.11 , confidence = 0.57 , similarity
    = 0.0000
Rule 28:   {('share/portion#2' , 0) } -> {('iraq/nnp' , 2) } , support = 0.17 ,
    confidence = 0.50 , similarity = 0.0000
Rule 29:   {('expectation/outlook#1' , 0) } -> {('iraq/nnp' , 2) } , support =
    0.11 , confidence = 0.67 , similarity = 0.0000
Rule 30:   {('country/state#1' , 0) ('consequence/effect#1' , 1) } -> {('iraq/nnp'
     , 2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 31:   {('people#1' , 0) } -> {('iraq/nnp' , 2) } , support = 0.17 ,
    confidence = 0.67 , similarity = 0.0000
Rule 32:   {('universe/existence#1' , 0) } -> {('iraq/nnp' , 2) } , support =
    0.14 , confidence = 0.71 , similarity = 0.0000
Rule 33:   {('datum/data_point#1' , 0) } -> {('iraq/nnp' , 2) } , support = 0.17
    , confidence = 0.60 , similarity = 0.0000
Rule 34:   {('year#3' , 0) } -> {('iraq/nnp' , 2) } , support = 0.19 ,
    confidence = 0.50 , similarity = 0.0000
Rule 35:   {('iran/nnp' , 0) } -> {('iraq/nnp' , 2) } , support = 0.11 ,
    confidence = 0.57 , similarity = 0.0000
Rule 36:   {('share/portion#2' 'datum/data_point#1' , 0) } -> {('iraq/nnp' , 2) }
    , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 37:   {('eu/nnp' 'europe/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 38:   {('iraq/nnp' 'europe/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 , confidence = 0.80 , similarity = 0.0000
Rule 39:   {('law/jurisprudence#2' , 1) ('year#3' , 0) } -> {('eu/nnp' , 2) } ,
    support = 0.14 , confidence = 0.63 , similarity = 0.0000
Rule 40:   {('law/jurisprudence#2' , 1) ('consequence/effect#1' , 0) } -> {('eu/
    nnp' , 2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 41:   {('law/jurisprudence#2' , 1) ('country/state#1' , 0) } -> {('eu/nnp' ,
    2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 42:   {('law/jurisprudence#2' , 1) ('euro#1' , 0) } -> {('eu/nnp' , 2) } ,
    support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 43:   {('law/jurisprudence#2' , 1) ('iraq/nnp' , 0) } -> {('eu/nnp' , 2) } ,
     support = 0.11 , confidence = 0.80 , similarity = 0.0000
Rule 44:   {('law/jurisprudence#2' , 1) ('europe/nnp' , 0) } -> {('eu/nnp' , 2) }
    , support = 0.14 , confidence = 0.71 , similarity = 0.0000
Rule 45:   {('eu/nnp' 'iraq/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 46:   {('europe/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } , support =
    0.19 , confidence = 0.54 , similarity = 0.0000
Rule 47:   {('dollar/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } , support =
    0.11 , confidence = 0.50 , similarity = 0.0000
Rule 48:   {('bush/nnp' 'iraq/nnp' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.14 , confidence = 0.83 ,
    similarity = 0.0000
Rule 49:   {('iraq/nnp' 'country/state#1' , 0) } -> {('bush/nnp' , 1) } , support
     = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 50:   {('plan/program#1' , 0) } -> {('bush/nnp' , 1) } , support = 0.14 ,
    confidence = 0.63 , similarity = 0.0000
Rule 51:   {('iraq/nnp' , 0) } -> {('economy/economic_system#1' , 1) } , support
   = 0.22 , confidence = 0.53 , similarity = 0.0000
Rule 52:   {('eu/nnp' 'iraq/nnp' , 0) } -> {('europe/nnp' , 1) } , support = 0.11
     , confidence = 0.50 , similarity = 0.0000
Rule 53:   {('country/state#1' 'consequence/effect#1' , 0) } -> {('europe/nnp' ,
    1) } , support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 54:   {('datum/data_point#1' , 0) } -> {('europe/nnp' , 1) } , support =
    0.14 , confidence = 0.50 , similarity = 0.0000
Rule 55:   {('reform#1' 'country/state#1' , 0) } -> {('europe/nnp' , 1) } ,
    support = 0.11 , confidence = 0.57 , similarity = 0.0000
```

```
Rule 56:  {('eu/nnp' 'iraq/nnp' , 0) } -> {('consequence/effect#1' , 1) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 57:  {('eu/nnp' , 0) } -> {('economy/economic_system#1' , 1) } , support =
    0.19 , confidence = 0.50 , similarity = 0.0000
Rule 58:  {('eu/nnp' , 0) } -> {('commercial_enterprise/business_enterprise#2' ,
    1) } , support = 0.22 , confidence = 0.57 , similarity = 0.0000
Rule 59:  {('eu/nnp' , 0) ('market/marketplace#1' , 1) } -> {('iraq/nnp' , 2) } ,
     support = 0.11 , confidence = 0.80 , similarity = 0.0000
Rule 60:  {('eu/nnp' , 0) } -> {('year#3' , 1) } , support = 0.25 , confidence
    = 0.64 , similarity = 0.0000
Rule 61:  {('law/jurisprudence#2' 'europe/nnp' , 0) } -> {('eu/nnp' , 1) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 62:  {('euro#1' , 0) } -> {('eu/nnp' , 1) } , support = 0.17 , confidence
    = 0.55 , similarity = 0.0000
Rule 63:  {('dollar#1' , 0) } -> {('eu/nnp' , 1) } , support = 0.14 ,
    confidence = 0.50 , similarity = 0.0000
Rule 64:  {('market/marketplace#1' , 0) } -> {('eu/nnp' , 1) } , support = 0.17
    , confidence = 0.55 , similarity = 0.0000
Rule 65:  {('share/portion#2' , 0) } -> {('eu/nnp' , 1) } , support = 0.22 ,
    confidence = 0.67 , similarity = 0.0000
Rule 66:  {('president_of_the_united_states/united_states_president#1' , 0) } ->
    {('eu/nnp' , 1) } , support = 0.14 , confidence = 0.56 , similarity =
    0.0000
Rule 67:  {('eu/nnp' , 1) ('economy/economic_system#1' , 0) } -> {('iraq/nnp' ,
    2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 68:  {('eu/nnp' , 1) ('share/portion#2' , 0) } -> {('iraq/nnp' , 2) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 69:  {('biology/biological_science#1' , 0) } -> {('eu/nnp' , 1) } , support
     = 0.14 , confidence = 0.50 , similarity = 0.0000
Rule 70:  {('economy/economic_system#1' , 0) } -> {('eu/nnp' , 1) } , support =
    0.19 , confidence = 0.50 , similarity = 0.0000
Rule 71:  {('market/marketplace#1' 'economy/economic_system#1' , 0) } -> {('eu/
    nnp' , 1) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 72:  {('year#3' , 0) } -> {('eu/nnp' , 1) } , support = 0.19 , confidence
    = 0.50 , similarity = 0.0000
Rule 73:  {('share/portion#2' 'economy/economic_system#1' , 0) } -> {('eu/nnp' ,
    1) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 74:  {('share/portion#2' 'datum/data_point#1' , 0) } -> {('eu/nnp' , 1) } ,
     support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 75:  {('dollar/nnp' , 0) } -> {('eu/nnp' , 1) } , support = 0.11 ,
    confidence = 0.50 , similarity = 0.0000
Rule 76:  {('iraq/nnp' 'economy/economic_system#1' , 0) } -> {('eu/nnp' , 1) } ,
     support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 77:  {('eu/nnp' , 1) ('iraq/nnp' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 2) } , support = 0.11 , confidence = 0.80 ,
    similarity = 0.0000
Rule 78:  {('law/jurisprudence#2' 'consequence/effect#1' , 0) } -> {('eu/nnp' ,
    2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 79:  {('country/state#1' 'depository_financial_institution/bank#1' , 0) } ->
     {('eu/nnp' , 2) } , support = 0.14 , confidence = 0.83 , similarity =
    0.0000
Rule 80:  {('iraq/nnp' 'euro#1' , 0) } -> {('eu/nnp' , 2) } , support = 0.14 ,
    confidence = 0.83 , similarity = 0.0000
Rule 81:  {('iraq/nnp' 'profit/gain#2' , 0) } -> {('eu/nnp' , 2) } , support =
    0.11 , confidence = 0.80 , similarity = 0.0000
Rule 82:  {('year#3' 'country/state#1' , 0) } -> {('eu/nnp' , 2) } , support =
    0.11 , confidence = 0.50 , similarity = 0.0000
Rule 83:  {('plan/program#1' , 0) } -> {('eu/nnp' , 2) } , support = 0.14 ,
    confidence = 0.63 , similarity = 0.0000
Rule 84:  {('law/jurisprudence#2' 'iraq/nnp' , 0) } -> {('eu/nnp' , 2) } ,
    support = 0.11 , confidence = 0.57 , similarity = 0.0000
```

```
Rule 85:  {('europe/nnp' 'company#1' , 0) } -> {('eu/nnp' , 2) } , support =
   0.11 , confidence = 0.57 , similarity = 0.0000
Rule 86:  {('country/state#1' , 0) } -> {('eu/nnp' , 2) } , support = 0.22 ,
   confidence = 0.57 , similarity = 0.0000
Rule 87:  {('country/state#1' , 0) ('consequence/effect#1' , 1) } -> {('eu/nnp' ,
    2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 88:  {('iraq/nnp' 'country/state#1' , 0) } -> {('eu/nnp' , 2) } , support =
    0.17 , confidence = 0.75 , similarity = 0.0000
Rule 89:  {('bush/nnp' , 0) } -> {('eu/nnp' , 2) } , support = 0.17 ,
   confidence = 0.67 , similarity = 0.0000
Rule 90:  {('law/jurisprudence#2' 'company#1' , 0) } -> {('eu/nnp' , 2) } ,
   support = 0.11 , confidence = 0.67 , similarity = 0.0000
Rule 91:  {('company#1' , 0) } -> {('eu/nnp' , 2) } , support = 0.17 ,
   confidence = 0.55 , similarity = 0.0000
Rule 92:  {('company#1' , 0) ('government/governing#3' , 1) } -> {('eu/nnp' , 2)
   } , support = 0.11 , confidence = 0.80 , similarity = 0.0000
Rule 93:  {('year#3' , 0) } -> {('eu/nnp' , 2) } , support = 0.19 , confidence
   = 0.50 , similarity = 0.0000
Rule 94:  {('law/jurisprudence#2' 'country/state#1' , 0) } -> {('eu/nnp' , 2) } ,
     support = 0.14 , confidence = 0.71 , similarity = 0.0000
Rule 95:  {('iraq/nnp' 'europe/nnp' , 0) } -> {('eu/nnp' , 2) } , support = 0.11
    , confidence = 0.80 , similarity = 0.0000
Rule 96:  {('depository_financial_institution/bank#1' , 0) } -> {('eu/nnp' , 2) }
    , support = 0.17 , confidence = 0.75 , similarity = 0.0000
Rule 97:  {('yukos/nnp' , 0) } -> {('eu/nnp' , 2) } , support = 0.11 ,
   confidence = 0.57 , similarity = 0.0000
Rule 98:  {('year#3' , 1) ('country/state#1' , 0) } -> {('eu/nnp' , 2) } ,
   support = 0.14 , confidence = 0.71 , similarity = 0.0000
Rule 99:  {('iraq/nnp' , 0) } -> {('eu/nnp' , 2) } , support = 0.25 ,
   confidence = 0.60 , similarity = 0.0000
Rule 100:  {('iraq/nnp' , 0) ('europe/nnp' , 1) } -> {('eu/nnp' , 2) } , support
    = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 101:  {('iraq/nnp' , 0) ('euro#1' , 1) } -> {('eu/nnp' , 2) } , support =
   0.14 , confidence = 1.00 , similarity = 0.0000
Rule 102:  {('iraq/nnp' , 0) ('commercial_enterprise/business_enterprise#2' , 1)
   } -> {('eu/nnp' , 2) } , support = 0.11 , confidence = 0.80 , similarity =
   0.0000
Rule 103:  {('iraq/nnp' , 0) ('share/portion#2' , 1) } -> {('eu/nnp' , 2) } ,
   support = 0.14 , confidence = 0.83 , similarity = 0.0000
Rule 104:  {('iraq/nnp' , 0) ('consequence/effect#1' , 1) } -> {('eu/nnp' , 2) }
   , support = 0.14 , confidence = 0.71 , similarity = 0.0000
Rule 105:  {('iraq/nnp' , 0) ('economy/economic_system#1' , 1) } -> {('eu/nnp' ,
   2) } , support = 0.17 , confidence = 0.75 , similarity = 0.0000
Rule 106:  {('iraq/nnp' 'plan/program#1' , 0) } -> {('eu/nnp' , 2) } , support =
    0.11 , confidence = 0.80 , similarity = 0.0000
Rule 107:  {('law/jurisprudence#2' 'europe/nnp' , 0) } -> {('eu/nnp' , 2) } ,
   support = 0.11 , confidence = 0.50 , similarity = 0.0000
Rule 108:  {('dollar#1' , 0) } -> {('eu/nnp' , 2) } , support = 0.14 ,
   confidence = 0.50 , similarity = 0.0000
Rule 109:  {('iraq/nnp' , 1) ('commercial_enterprise/business_enterprise#2' , 0)
   } -> {('eu/nnp' , 2) } , support = 0.11 , confidence = 0.67 , similarity =
   0.0000
Rule 110:  {('europe/nnp' , 1) ('country/state#1' , 0) } -> {('eu/nnp' , 2) } ,
   support = 0.11 , confidence = 0.67 , similarity = 0.0000
Rule 111:  {('commercial_enterprise/business_enterprise#2' , 1) ('company#1' , 0)
    } -> {('eu/nnp' , 2) } , support = 0.14 , confidence = 0.71 , similarity =
    0.0000
Rule 112:  {('law/jurisprudence#2' 'year#3' , 0) } -> {('eu/nnp' , 2) } ,
   support = 0.11 , confidence = 0.57 , similarity = 0.0000
Rule 113:  {('profit/gain#2' , 0) } -> {('eu/nnp' , 2) } , support = 0.17 ,
   confidence = 0.86 , similarity = 0.0000
```

```
Rule 114:   {('commercial_enterprise/business_enterprise#2' 'company#1' , 0) } ->
    {('eu/nnp' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,  similarity =
    0.0000
Rule 115:   {('iraq/nnp' 'company#1' , 0) } -> {('eu/nnp' , 2) } ,  support = 0.11
     ,  confidence = 0.80 ,  similarity = 0.0000
Rule 116:   {('year#3' 'europe/nnp' , 0) } -> {('eu/nnp' , 2) } ,  support = 0.14
    ,  confidence = 0.63 ,  similarity = 0.0000
Rule 117:   {('occupation/business#1' , 0) } -> {('eu/nnp' , 2) } ,  support =
    0.11 ,  confidence = 0.50 ,  similarity = 0.0000
Rule 118:   {('company#1' 'country/state#1' , 0) } -> {('eu/nnp' , 2) } ,  support
     = 0.11 ,  confidence = 0.57 ,  similarity = 0.0000
Rule 119:   {('iraq/nnp' 'commercial_enterprise/business_enterprise#2' , 0) } ->
    {('eu/nnp' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity =
    0.0000
Rule 120:   {('dollar/nnp' , 0) } -> {('eu/nnp' , 2) } ,  support = 0.11 ,
    confidence = 0.50 ,  similarity = 0.0000
Rule 121:   {('country/state#1' 'government/governing#3' , 0) } -> {('eu/nnp' , 2)
     } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0000
Rule 122:   {('dollar/nnp' , 0) } -> {('consequence/effect#1' , 2) } ,  support =
    0.11 ,  confidence = 0.50 ,  similarity = 0.0000
Rule 123:   {('dollar/nnp' , 0) } -> {('country/state#1' , 1) } ,  support = 0.11
    ,  confidence = 0.50 ,  similarity = 0.0000
Rule 124:   {('dollar/nnp' , 0) } -> {('dollar#1' , 1) } ,  support = 0.11 ,
    confidence = 0.50 ,  similarity = 0.0000
Rule 125:   {('eu/nnp' 'europe/nnp' , 0) } -> {('year#3' , 1) } ,  support = 0.14
    ,  confidence = 0.63 ,  similarity = 0.0000
Rule 126:   {('iraq/nnp' , 0) ('europe/nnp' , 1) } -> {('law/jurisprudence#2' , 2)
     } ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0000
Rule 127:   {('iran/nnp' , 0) } -> {('law/jurisprudence#2' , 2) } ,  support =
    0.14 ,  confidence = 0.71 ,  similarity = 0.0000
Rule 128:   {('yen#2' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support = 0.11 ,
      confidence = 0.57 ,  similarity = 0.0485
Rule 129:   {('depository_financial_institution/bank#1' , 0) } -> {('dollar#1' ,
    2) } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0494
Rule 130:   {('quarter#6' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support =
    0.11 ,  confidence = 0.67 ,  similarity = 0.0509
Rule 131:   {('iraq/nnp' 'euro#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 ,  confidence = 0.67 ,  similarity = 0.0524
Rule 132:   {('euro#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support = 0.19
    ,  confidence = 0.64 ,  similarity = 0.0524
Rule 133:   {('euro#1' 'europe/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.14 ,  confidence = 0.71 ,  similarity = 0.0524
Rule 134:   {('euro#1' 'europe/nnp' , 0) } -> {('law/jurisprudence#2' , 2) } ,
    support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0524
Rule 135:   {('europe/nnp' 'company#1' , 0) } -> {('euro#1' , 1) } ,  support =
    0.14 ,  confidence = 0.71 ,  similarity = 0.0533
Rule 136:   {('quarter#6' , 0) } -> {('economy/economic_system#1' , 2) } ,
    support = 0.11 ,  confidence = 0.67 ,  similarity = 0.0536
Rule 137:   {('reform#1' 'euro#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 ,  confidence = 1.00 ,  similarity = 0.0537
Rule 138:   {('school_term/academic_term#1' , 0) } -> {('law/jurisprudence#2' , 2)
     } ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0543
Rule 139:   {('market/marketplace#1' 'quarter#6' , 0) } -> {('law/jurisprudence#2'
     , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity = 0.0549
Rule 140:   {('iraq/nnp' 'euro#1' , 0) } -> {('economy/economic_system#1' , 1) } ,
      support = 0.11 ,  confidence = 0.67 ,  similarity = 0.0551
Rule 141:   {('reform#1' 'eu/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 ,  confidence = 0.80 ,  similarity = 0.0551
Rule 142:   {('reform#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support =
    0.19 ,  confidence = 0.78 ,  similarity = 0.0551
```

```
Rule 143:   {('law/jurisprudence#2' , 1) ('euro#1' , 0) } -> {('market/marketplace
    #1' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0555
Rule 144:   {('percentage/percent#1' , 0) } -> {('market/marketplace#1' , 1) } ,
    support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0557
Rule 145:   {('biology/biological_science#1' , 0) } -> {('law/jurisprudence#2' ,
    1) } ,  support = 0.14 ,  confidence = 0.50 ,  similarity = 0.0560
Rule 146:   {('dollar#1' , 0) } -> {('share/portion#2' , 2) } ,  support = 0.14 ,
     confidence = 0.50 ,  similarity = 0.0568
Rule 147:   {('euro#1' 'government/governing#3' , 0) } -> {('law/jurisprudence#2'
    , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity = 0.0570
Rule 148:   {('sale/cut-rate_sale#4' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.14 ,  confidence = 0.56 ,  similarity = 0.0572
Rule 149:   {('interest_rate/rate_of_interest#1' , 0) } -> {('economy/
    economic_system#1' , 1) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity
     = 0.0573
Rule 150:   {('euro#1' 'year#3' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.14 ,  confidence = 0.71 ,  similarity = 0.0578
Rule 151:   {('dollar#1' , 0) } -> {('commercial_enterprise/business_enterprise#2'
     , 2) } ,  support = 0.14 ,  confidence = 0.50 ,  similarity = 0.0580
Rule 152:   {('dollar#1' , 0) } -> {('commercial_enterprise/business_enterprise#2'
     , 1) } ,  support = 0.14 ,  confidence = 0.50 ,  similarity = 0.0580
Rule 153:   {('market/marketplace#1' 'quarter#6' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity
     = 0.0580
Rule 154:   {('day#4' 'euro#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support
     = 0.11 ,  confidence = 0.80 ,  similarity = 0.0584
Rule 155:   {('dollar#1' , 0) } -> {('percentage/percent#1' , 1) } ,  support =
    0.14 ,  confidence = 0.50 ,  similarity = 0.0587
Rule 156:   {('market/marketplace#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.17 ,  confidence = 0.55 ,  similarity = 0.0588
Rule 157:   {('plan/program#1' , 0) } -> {('dollar#1' , 1) } ,  support = 0.11 ,
    confidence = 0.50 ,  similarity = 0.0589
Rule 158:   {('policy#2' , 0) } -> {('market/marketplace#1' , 2) } ,  support =
    0.11 ,  confidence = 0.57 ,  similarity = 0.0589
Rule 159:   {('biology/biological_science#1' , 0) } -> {('economy/economic_system
    #1' , 2) } ,  support = 0.14 ,  confidence = 0.50 ,  similarity = 0.0592
Rule 160:   {('interest_rate/rate_of_interest#1' , 0) } -> {('consequence/effect
    #1' , 1) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0592
Rule 161:   {('policy#2' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support =
    0.11 ,  confidence = 0.57 ,  similarity = 0.0593
Rule 162:   {('depository_financial_institution/bank#1' , 0) } -> {('year#3' , 1)
    } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0593
Rule 163:   {('policy#2' , 0) } -> {('law/jurisprudence#2' , 2) } ,  support =
    0.11 ,  confidence = 0.57 ,  similarity = 0.0593
Rule 164:   {('biology/biological_science#1' , 0) } -> {('year#3' , 2) } ,
    support = 0.14 ,  confidence = 0.50 ,  similarity = 0.0597
Rule 165:   {('reform#1' , 0) } -> {('consequence/effect#1' , 1) } ,  support =
    0.14 ,  confidence = 0.56 ,  similarity = 0.0601
Rule 166:   {('week/hebdomad#1' , 0) } -> {('law/jurisprudence#2' , 2) } ,
    support = 0.17 ,  confidence = 0.67 ,  similarity = 0.0602
Rule 167:   {('eu/nnp' 'company#1' , 0) } -> {('report/study#1' , 2) } ,  support
    = 0.11 ,  confidence = 0.57 ,  similarity = 0.0603
Rule 168:   {('sale/cut-rate_sale#4' , 0) } -> {('economy/economic_system#1' , 2)
    } ,  support = 0.14 ,  confidence = 0.56 ,  similarity = 0.0605
Rule 169:   {('law/jurisprudence#2' 'economy/economic_system#1' , 0) } -> {('
    market/marketplace#1' , 1) } ,  support = 0.11 ,  confidence = 0.67 ,
    similarity = 0.0606
Rule 170:   {('interest_rate/rate_of_interest#1' , 0) } -> {('
    commercial_enterprise/business_enterprise#2' , 1) } ,  support = 0.11 ,
    confidence = 0.57 ,  similarity = 0.0607
```

```
Rule 171:   {('law/jurisprudence#2' 'euro#1' , 0) } -> {('share/portion#2' , 2) }
    , support = 0.11 , confidence = 0.50 , similarity = 0.0611
Rule 172:   {('biology/biological_science#1' , 0) } -> {('consequence/effect#1' ,
    1) } , support = 0.14 , confidence = 0.50 , similarity = 0.0612
Rule 173:   {('president_of_the_united_states/united_states_president#1' , 0) } ->
     {('market/marketplace#1' , 2) } , support = 0.14 , confidence = 0.56 ,
    similarity = 0.0613
Rule 174:   {('biology/biological_science#1' , 0) } -> {('share/portion#2' , 1) }
    , support = 0.14 , confidence = 0.50 , similarity = 0.0614
Rule 175:   {('populace/public#1' , 0) } -> {('market/marketplace#1' , 2) } ,
    support = 0.11 , confidence = 0.67 , similarity = 0.0616
Rule 176:   {('government/governing#3' , 0) } -> {('law/jurisprudence#2' , 1) } ,
     support = 0.14 , confidence = 0.50 , similarity = 0.0616
Rule 177:   {('president_of_the_united_states/united_states_president#1' , 0) } ->
     {('law/jurisprudence#2' , 1) } , support = 0.17 , confidence = 0.67 ,
    similarity = 0.0617
Rule 178:   {('week/hebdomad#1' , 0) } -> {('law/jurisprudence#2' 'economy/
    economic_system#1' , 2) } , support = 0.14 , confidence = 0.56 , similarity
     = 0.0621
Rule 179:   {('market/marketplace#1' , 0) } -> {('economy/economic_system#1' , 2)
    } , support = 0.17 , confidence = 0.55 , similarity = 0.0624
Rule 180:   {('market/marketplace#1' , 0) } -> {('economy/economic_system#1' , 1)
    } , support = 0.17 , confidence = 0.55 , similarity = 0.0624
Rule 181:   {('eu/nnp' , 1) ('market/marketplace#1' , 0) } -> {('economy/
    economic_system#1' , 2) } , support = 0.14 , confidence = 0.83 , similarity
     = 0.0624
Rule 182:   {('dollar#1' , 0) } -> {('country/state#1' , 1) } , support = 0.14 ,
     confidence = 0.50 , similarity = 0.0625
Rule 183:   {('biology/biological_science#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 2) } , support = 0.14 , confidence = 0.50 ,
    similarity = 0.0628
Rule 184:   {('biology/biological_science#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.17 , confidence = 0.60 ,
    similarity = 0.0628
Rule 185:   {('eu/nnp' 'year#3' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 , confidence = 0.80 , similarity = 0.0633
Rule 186:   {('year#3' , 0) } -> {('law/jurisprudence#2' , 1) } , support = 0.22
    , confidence = 0.57 , similarity = 0.0633
Rule 187:   {('year#3' 'europe/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.17 , confidence = 0.75 , similarity = 0.0633
Rule 188:   {('expectation/outlook#1' , 0) } -> {('economy/economic_system#1' , 2)
     } , support = 0.11 , confidence = 0.67 , similarity = 0.0633
Rule 189:   {('law/jurisprudence#2' 'eu/nnp' , 0) } -> {('year#3' , 1) } ,
    support = 0.17 , confidence = 0.75 , similarity = 0.0633
Rule 190:   {('law/jurisprudence#2' 'europe/nnp' , 0) } -> {('year#3' , 1) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.0633
Rule 191:   {('year#3' , 1) ('europe/nnp' , 0) } -> {('law/jurisprudence#2' , 2) }
     , support = 0.11 , confidence = 0.67 , similarity = 0.0633
Rule 192:   {('president_of_the_united_states/united_states_president#1' , 0) } ->
     {('law/jurisprudence#2' , 1) ('economy/economic_system#1' , 2) } , support =
     0.14 , confidence = 0.56 , similarity = 0.0636
Rule 193:   {('day#4' 'year#3' , 0) } -> {('law/jurisprudence#2' , 1) } , support
     = 0.11 , confidence = 0.67 , similarity = 0.0638
Rule 194:   {('reform#1' 'country/state#1' , 0) } -> {('law/jurisprudence#2' , 1)
    } , support = 0.14 , confidence = 0.71 , similarity = 0.0639
Rule 195:   {('week/hebdomad#1' , 0) } -> {('economy/economic_system#1' , 2) } ,
    support = 0.14 , confidence = 0.56 , similarity = 0.0639
Rule 196:   {('consequence/effect#1' 'biology/biological_science#1' , 0) } -> {('
    economy/economic_system#1' , 1) } , support = 0.11 , confidence = 0.67 ,
    similarity = 0.0642
```

```
Rule 197:   {('iraq/nnp' 'universe/existence#1' , 0) } -> {('economy/
    economic_system#1' , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity
     = 0.0643
Rule 198:   {('iraq/nnp' 'universe/existence#1' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity
     = 0.0643
Rule 199:   {('universe/existence#1' , 0) } -> {('economy/economic_system#1' , 2)
    } ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0643
Rule 200:   {('universe/existence#1' , 0) } -> {('economy/economic_system#1' , 1)
    } ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0643
Rule 201:   {('sale/cut-rate-sale#4' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } ,  support = 0.14 ,  confidence = 0.56 ,
    similarity = 0.0643
Rule 202:   {('day#4' , 0) } -> {('law/jurisprudence#2' , 1) } ,  support = 0.14 ,
      confidence = 0.71 ,  similarity = 0.0644
Rule 203:   {('market/marketplace#1' , 0) } -> {('consequence/effect#1' , 1) } ,
    support = 0.17 ,  confidence = 0.55 ,  similarity = 0.0646
Rule 204:   {('europe/nnp' 'company#1' , 0) } -> {('year#3' , 1) } ,  support =
    0.11 ,  confidence = 0.57 ,  similarity = 0.0646
Rule 205:   {('currency#1' 'government/governing#3' , 0) } -> {('law/jurisprudence
    #2' , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity = 0.0648
Rule 206:   {('consequence/effect#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.19 ,  confidence = 0.50 ,  similarity = 0.0650
Rule 207:   {('law/jurisprudence#2' 'eu/nnp' , 0) } -> {('consequence/effect#1' ,
    1) } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0650
Rule 208:   {('iraq/nnp' , 0) ('consequence/effect#1' , 1) } -> {('law/
    jurisprudence#2' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity =
     0.0650
Rule 209:   {('report/study#1' , 0) } -> {('consequence/effect#1' , 1) } ,
    support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0650
Rule 210:   {('policy#2' , 0) } -> {('consequence/effect#1' , 1) } ,  support =
    0.11 ,  confidence = 0.57 ,  similarity = 0.0651
Rule 211:   {('law/jurisprudence#2' 'eu/nnp' , 0) } -> {('share/portion#2' , 1) }
    ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0652
Rule 212:   {('share/portion#2' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.17 ,  confidence = 0.50 ,  similarity = 0.0652
Rule 213:   {('government/governing#3' , 0) } -> {('economy/economic_system#1' ,
    1) } ,  support = 0.14 ,  confidence = 0.50 ,  similarity = 0.0655
Rule 214:   {('president_of_the_united_states/united_states_president#1' , 0) } ->
     {('economy/economic_system#1' , 2) } ,  support = 0.19 ,  confidence = 0.78 ,
      similarity = 0.0655
Rule 215:   {('expectation/outlook#1' , 0) } -> {('consequence/effect#1' , 1) } ,
     support = 0.14 ,  confidence = 0.83 ,  similarity = 0.0656
Rule 216:   {('consequence/effect#1' 'market/marketplace#1' , 0) } -> {('economy/
    economic_system#1' , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity
     = 0.0658
Rule 217:   {('consequence/effect#1' , 1) ('market/marketplace#1' , 0) } -> {('
    economy/economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,
    similarity = 0.0658
Rule 218:   {('share/portion#2' 'market/marketplace#1' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,  similarity
     = 0.0659
Rule 219:   {('report/study#1' 'government/governing#3' , 0) } -> {('consequence/
    effect#1' , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity = 0.0664
Rule 220:   {('company#1' , 0) } -> {('consequence/effect#1' , 2) } ,  support =
    0.17 ,  confidence = 0.55 ,  similarity = 0.0664
Rule 221:   {('law/jurisprudence#2' 'eu/nnp' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } ,  support = 0.11 ,  confidence = 0.50 ,
    similarity = 0.0668
Rule 222:   {('consequence/effect#1' 'economy/economic_system#1' , 0) } -> {('
    president_of_the_united_states/united_states_president#1' , 1) } ,  support =
    0.11 , confidence = 0.50 , similarity = 0.0668
Rule 223: {('market/marketplace#1' 'economy/economic_system#1' , 0) } -> {('
    consequence/effect#1' , 1) } , support = 0.14 , confidence = 0.71 ,
    similarity = 0.0669
Rule 224: {('minister/government_minister#2' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.11 , confidence = 0.67 ,
    similarity = 0.0671
Rule 225: {('country/state#1' 'government/governing#3' , 0) } -> {('law/
    jurisprudence#2' , 2) } , support = 0.11 , confidence = 0.50 , similarity =
     0.0672
Rule 226: {('year#3' 'europe/nnp' , 0) } -> {('economy/economic_system#1' , 1) }
     , support = 0.14 , confidence = 0.63 , similarity = 0.0674
Rule 227: {('president_of_the_united_states/united_states_president#1' '
    consequence/effect#1' , 0) } -> {('economy/economic_system#1' , 2) } ,
    support = 0.11 , confidence = 0.80 , similarity = 0.0674
Rule 228: {('year#3' , 1) ('europe/nnp' , 0) } -> {('economy/economic_system#1'
     , 2) } , support = 0.11 , confidence = 0.67 , similarity = 0.0674
Rule 229: {('law/jurisprudence#2' 'company#1' , 0) } -> {('commercial_enterprise
    /business_enterprise#2' , 1) } , support = 0.11 , confidence = 0.67 ,
    similarity = 0.0675
Rule 230: {('depository_financial_institution/bank#1' , 0) } -> {('country/state
    #1' , 2) } , support = 0.11 , confidence = 0.50 , similarity = 0.0676
Rule 231: {('monetary_unit#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 , confidence = 0.57 , similarity = 0.0678
Rule 232: {('currency#1' , 0) } -> {('law/jurisprudence#2' , 1) } , support =
    0.14 , confidence = 0.83 , similarity = 0.0679
Rule 233: {('government/governing#3' , 0) } -> {('consequence/effect#1' , 1) } ,
     support = 0.14 , confidence = 0.50 , similarity = 0.0679
Rule 234: {('president_of_the_united_states/united_states_president#1' , 0) } ->
     {('consequence/effect#1' , 2) } , support = 0.14 , confidence = 0.56 ,
    similarity = 0.0680
Rule 235: {('year#3' 'country/state#1' , 0) } -> {('law/jurisprudence#2' , 1) }
     , support = 0.14 , confidence = 0.63 , similarity = 0.0681
Rule 236: {('share/portion#2' , 0) } -> {('government/governing#3' , 1) } ,
    support = 0.17 , confidence = 0.50 , similarity = 0.0681
Rule 237: {('year#3' , 1) ('country/state#1' , 0) } -> {('law/jurisprudence#2' ,
     2) } , support = 0.11 , confidence = 0.57 , similarity = 0.0681
Rule 238: {('year#3' 'consequence/effect#1' , 0) } -> {('economy/economic_system
    #1' , 1) } , support = 0.14 , confidence = 0.83 , similarity = 0.0683
Rule 239: {('bush/nnp' 'company#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.11 , confidence = 1.00 ,
    similarity = 0.0683
Rule 240: {('europe/nnp' 'company#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.11 , confidence = 0.57 ,
    similarity = 0.0683
Rule 241: {('eu/nnp' 'company#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.14 , confidence = 0.71 ,
    similarity = 0.0683
Rule 242: {('company#1' , 0) } -> {('commercial_enterprise/business_enterprise
    #2' , 1) } , support = 0.19 , confidence = 0.64 , similarity = 0.0683
Rule 243: {('occupation/business#1' 'market/marketplace#1' , 0) } -> {('economy/
    economic_system#1' , 1) } , support = 0.14 , confidence = 1.00 , similarity
     = 0.0685
Rule 244: {('occupation/business#1' 'market/marketplace#1' , 0) } -> {('economy/
    economic_system#1' , 2) } , support = 0.11 , confidence = 0.80 , similarity
     = 0.0685
Rule 245: {('occupation/business#1' , 0) ('market/marketplace#1' , 1) } -> {('
    economy/economic_system#1' , 2) } , support = 0.11 , confidence = 0.80 ,
    similarity = 0.0685
Rule 246: {('country/state#1' 'consequence/effect#1' , 0) } -> {('law/
    jurisprudence#2' , 1) } , support = 0.11 , confidence = 0.50 , similarity =
     0.0689
Rule 247:   {('reform#1' 'country/state#1' , 0) } -> {('year#3' , 1) } ,  support
    = 0.11 ,  confidence = 0.57 ,  similarity = 0.0689
Rule 248:   {('country/state#1' , 0) ('consequence/effect#1' , 1) } -> {('law/
    jurisprudence#2' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity =
     0.0689
Rule 249:   {('iraq/nnp' , 1) ('consequence/effect#1' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,  similarity
     = 0.0693
Rule 250:   {('bush/nnp' 'consequence/effect#1' , 0) } -> {('economy/
    economic_system#1' , 1) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity
     = 0.0693
Rule 251:   {('iraq/nnp' 'consequence/effect#1' , 0) } -> {('economy/
    economic_system#1' , 1) } ,  support = 0.11 ,  confidence = 0.67 ,  similarity
     = 0.0693
Rule 252:   {('consequence/effect#1' , 0) } -> {('economy/economic_system#1' , 1)
    } ,  support = 0.22 ,  confidence = 0.57 ,  similarity = 0.0693
Rule 253:   {('iraq/nnp' , 0) ('consequence/effect#1' , 1) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity
     = 0.0693
Rule 254:   {('iraq/nnp' , 1) ('share/portion#2' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.80 ,  similarity
     = 0.0695
Rule 255:   {('share/portion#2' , 0) } -> {('economy/economic_system#1' , 2) } ,
    support = 0.19 ,  confidence = 0.58 ,  similarity = 0.0695
Rule 256:   {('eu/nnp' , 1) ('share/portion#2' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.17 ,  confidence = 0.75 ,  similarity
     = 0.0695
Rule 257:   {('share/portion#2' , 0) } -> {('eu/nnp' , 1) ('economy/
    economic_system#1' , 2) } ,  support = 0.17 ,  confidence = 0.50 ,  similarity
     = 0.0695
Rule 258:   {('consequence/effect#1' , 0) } -> {('year#3' , 1) } ,  support = 0.19
     ,  confidence = 0.50 ,  similarity = 0.0700
Rule 259:   {('year#3' 'company#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } ,  support = 0.11 ,  confidence = 0.67 ,
    similarity = 0.0702
Rule 260:   {('share/portion#2' , 0) } -> {('year#3' , 1) } ,  support = 0.17 ,
    confidence = 0.50 ,  similarity = 0.0702
Rule 261:   {('commercial_enterprise/business_enterprise#2' 'consequence/effect#1'
     , 0) } -> {('economy/economic_system#1' , 1) } ,  support = 0.14 ,
    confidence = 0.71 ,  similarity = 0.0703
Rule 262:   {('commercial_enterprise/business_enterprise#2' , 1) ('company#1' , 0)
     } -> {('consequence/effect#1' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,
      similarity = 0.0703
Rule 263:   {('commercial_enterprise/business_enterprise#2' , 1) ('share/portion
    #2' , 0) } -> {('economy/economic_system#1' , 2) } ,  support = 0.11 ,
    confidence = 1.00 ,  similarity = 0.0704
Rule 264:   {('currency#1' , 0) } -> {('government/governing#3' , 1) } ,  support
    = 0.11 ,  confidence = 0.67 ,  similarity = 0.0711
Rule 265:   {('law/jurisprudence#2' 'country/state#1' , 0) } -> {('year#3' , 1) }
    ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0712
Rule 266:   {('iraq/nnp' 'commercial_enterprise/business_enterprise#2' , 0) } ->
     {('economy/economic_system#1' , 1) } ,  support = 0.11 ,  confidence = 0.57 ,
      similarity = 0.0714
Rule 267:   {('iraq/nnp' , 0) ('economy/economic_system#1' , 1) } -> {('
    commercial_enterprise/business_enterprise#2' , 2) } ,  support = 0.14 ,
    confidence = 0.63 ,  similarity = 0.0714
Rule 268:   {('computer_science/computing#1' , 0) } -> {('country/state#1' , 1) }
    ,  support = 0.11 ,  confidence = 0.57 ,  similarity = 0.0716
Rule 269:   {('share/portion#2' 'datum/data_point#1' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity
    = 0.0717
Rule 270:  {('company#1' 'country/state#1' , 0) } -> {('year#3' , 1) } , support
    = 0.11 , confidence = 0.57 , similarity = 0.0719
Rule 271:  {('year#3' 'europe/nnp' , 0) } -> {('commercial_enterprise/
   business_enterprise#2' , 1) } , support = 0.11 , confidence = 0.50 ,
   similarity = 0.0721
Rule 272:  {('consequence/effect#1' , 0) } -> {('share/portion#2' , 1) } ,
   support = 0.22 , confidence = 0.57 , similarity = 0.0723
Rule 273:  {('euro#1' 'company#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
   support = 0.11 , confidence = 0.80 , similarity = 0.0726
Rule 274:  {('country/state#1' 'government/governing#3' , 0) } -> {('year#3' , 1)
    } , support = 0.11 , confidence = 0.50 , similarity = 0.0726
Rule 275:  {('law/jurisprudence#2' 'europe/nnp' , 0) } -> {('country/state#1' ,
   2) } , support = 0.11 , confidence = 0.50 , similarity = 0.0728
Rule 276:  {('eu/nnp' 'country/state#1' , 0) } -> {('law/jurisprudence#2' , 1) }
    , support = 0.14 , confidence = 0.63 , similarity = 0.0728
Rule 277:  {('country/state#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
   support = 0.19 , confidence = 0.50 , similarity = 0.0728
Rule 278:  {('europe/nnp' 'country/state#1' , 0) } -> {('law/jurisprudence#2' ,
   1) } , support = 0.14 , confidence = 0.83 , similarity = 0.0728
Rule 279:  {('consequence/effect#1' 'economy/economic_system#1' , 0) } -> {('
   commercial_enterprise/business_enterprise#2' , 1) } , support = 0.11 ,
   confidence = 0.50 , similarity = 0.0728
Rule 280:  {('country/state#1' , 0) } -> {('law/jurisprudence#2' , 2) } ,
   support = 0.19 , confidence = 0.50 , similarity = 0.0728
Rule 281:  {('iraq/nnp' 'country/state#1' , 0) } -> {('law/jurisprudence#2' , 2)
   } , support = 0.14 , confidence = 0.63 , similarity = 0.0728
Rule 282:  {('plan/program#1' 'company#1' , 0) } -> {('commercial_enterprise/
   business_enterprise#2' , 1) } , support = 0.11 , confidence = 0.80 ,
   similarity = 0.0732
Rule 283:  {('plan/program#1' , 0) } -> {('year#3' , 1) } , support = 0.14 ,
   confidence = 0.63 , similarity = 0.0734
Rule 284:  {('country/state#1' 'report/study#1' , 0) } -> {('consequence/effect
   #1' , 1) } , support = 0.11 , confidence = 0.67 , similarity = 0.0734
Rule 285:  {('country/state#1' 'policy#2' , 0) } -> {('consequence/effect#1' , 1)
    } , support = 0.11 , confidence = 0.80 , similarity = 0.0735
Rule 286:  {('datum/data_point#1' , 0) } -> {('economy/economic_system#1' , 2) }
    , support = 0.17 , confidence = 0.60 , similarity = 0.0739
Rule 287:  {('iraq/nnp' , 0) ('economy/economic_system#1' , 1) } -> {('datum/
   data_point#1' , 2) } , support = 0.11 , confidence = 0.50 , similarity =
   0.0739
Rule 288:  {('consequence/effect#1' , 0) } -> {('commercial_enterprise/
   business_enterprise#2' , 1) } , support = 0.19 , confidence = 0.50 ,
   similarity = 0.0743
Rule 289:  {('eu/nnp' 'consequence/effect#1' , 0) } -> {('commercial_enterprise/
   business_enterprise#2' , 1) } , support = 0.11 , confidence = 0.67 ,
   similarity = 0.0743
Rule 290:  {('iraq/nnp' , 0) ('consequence/effect#1' , 1) } -> {('
   commercial_enterprise/business_enterprise#2' , 2) } , support = 0.11 ,
   confidence = 0.57 , similarity = 0.0743
Rule 291:  {('week/hebdomad#1' , 0) } -> {('country/state#1' , 2) } , support =
   0.17 , confidence = 0.67 , similarity = 0.0744
Rule 292:  {('occupation/business#1' , 0) } -> {('economy/economic_system#1' , 2)
    } , support = 0.17 , confidence = 0.75 , similarity = 0.0746
Rule 293:  {('occupation/business#1' , 0) } -> {('economy/economic_system#1' , 1)
    } , support = 0.14 , confidence = 0.63 , similarity = 0.0746
Rule 294:  {('country/state#1' 'consequence/effect#1' , 0) } -> {('year#3' , 1) }
    , support = 0.14 , confidence = 0.63 , similarity = 0.0746
Rule 295:  {('datum/data_point#1' , 0) } -> {('year#3' , 1) } , support = 0.19 ,
     confidence = 0.70 , similarity = 0.0747
```

```
Rule 296:  {('country/state#1' 'government/governing#3' , 0) } -> {('consequence/
    effect#1' , 1) } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0749
Rule 297:  {('share/portion#2' 'economy/economic_system#1' , 0) } -> {('datum/
    data_point#1' , 1) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity =
    0.0756
Rule 298:  {('law/jurisprudence#2' , 0) ('country/state#1' , 1) } -> {('
    commercial_enterprise/business_enterprise#2' , 2) } ,  support = 0.11 ,
    confidence = 0.80 ,  similarity = 0.0757
Rule 299:  {('law/jurisprudence#2' 'country/state#1' , 0) } -> {('
    commercial_enterprise/business_enterprise#2' , 1) } ,  support = 0.11 ,
    confidence = 0.57 ,  similarity = 0.0757
Rule 300:  {('plan/program#1' , 0) } -> {('share/portion#2' , 2) } ,  support =
    0.11 ,  confidence = 0.50 ,  similarity = 0.0760
Rule 301:  {('law/jurisprudence#2' , 1) ('year#3' , 0) } -> {('country/state#1' ,
     2) } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0760
Rule 302:  {('iraq/nnp' 'country/state#1' , 0) } -> {('government/governing#3' ,
    2) } ,  support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0765
Rule 303:  {('law/jurisprudence#2' , 1) ('euro#1' , 0) } -> {('economy/
    economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.57 ,  similarity
     = 0.0769
Rule 304:  {('occupation/business#1' , 0) } -> {('share/portion#2' , 1) } ,
    support = 0.11 ,  confidence = 0.50 ,  similarity = 0.0780
Rule 305:  {('year#3' 'company#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
    support = 0.11 ,  confidence = 0.67 ,  similarity = 0.0781
Rule 306:  {('plan/program#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } ,  support = 0.14 ,  confidence = 0.63 ,
    similarity = 0.0782
Rule 307:  {('economy/economic_system#1' , 0) } -> {('country/state#1' , 2) } ,
    support = 0.19 ,  confidence = 0.50 ,  similarity = 0.0783
Rule 308:  {('country/state#1' , 0) } -> {('economy/economic_system#1' , 1) } ,
    support = 0.19 ,  confidence = 0.50 ,  similarity = 0.0783
Rule 309:  {('year#3' 'country/state#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } ,  support = 0.11 ,  confidence = 0.50 ,
    similarity = 0.0784
Rule 310:  {('market/marketplace#1' 'economy/economic_system#1' , 0) } -> {('law/
    jurisprudence#2' , 1) } ,  support = 0.14 ,  confidence = 0.71 ,  similarity =
     0.0788
Rule 311:  {('music#1' , 0) } -> {('consequence/effect#1' , 1) } ,  support =
    0.14 ,  confidence = 0.83 ,  similarity = 0.0788
Rule 312:  {('eu/nnp' 'country/state#1' , 0) } -> {('year#3' , 1) } ,  support =
    0.14 ,  confidence = 0.63 ,  similarity = 0.0791
Rule 313:  {('country/state#1' , 0) } -> {('year#3' , 1) } ,  support = 0.19 ,
    confidence = 0.50 ,  similarity = 0.0791
Rule 314:  {('europe/nnp' 'country/state#1' , 0) } -> {('year#3' , 1) } ,
    support = 0.11 ,  confidence = 0.67 ,  similarity = 0.0791
Rule 315:  {('iraq/nnp' 'country/state#1' , 0) } -> {('year#3' , 1) } ,  support
   = 0.11 ,  confidence = 0.50 ,  similarity = 0.0791
Rule 316:  {('year#3' , 0) } -> {('country/state#1' , 2) } ,  support = 0.19 ,
    confidence = 0.50 ,  similarity = 0.0791
Rule 317:  {('country/state#1' 'consequence/effect#1' , 0) } -> {('
    commercial_enterprise/business_enterprise#2' , 1) } ,  support = 0.11 ,
    confidence = 0.50 ,  similarity = 0.0795
Rule 318:  {('week/hebdomad#1' 'economy/economic_system#1' , 0) } -> {('law/
    jurisprudence#2' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,  similarity =
     0.0795
Rule 319:  {('law/jurisprudence#2' , 0) ('market/marketplace#1' , 1) } -> {('
    economy/economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,
    similarity = 0.0805
Rule 320:  {('law/jurisprudence#2' , 1) ('market/marketplace#1' , 0) } -> {('
    economy/economic_system#1' , 2) } ,  support = 0.11 ,  confidence = 0.67 ,
    similarity = 0.0805
```

```
Rule 321:   {('eu/nnp' 'country/state#1' , 0) } -> {('consequence/effect#1' , 1) }
     , support = 0.11 , confidence = 0.50 , similarity = 0.0818
Rule 322:   {('country/state#1' , 0) } -> {('consequence/effect#1' , 1) } ,
    support = 0.19 , confidence = 0.50 , similarity = 0.0818
Rule 323:   {('law/jurisprudence#2' , 1) ('president_of_the_united_states/
    united_states_president#1' , 0) } -> {('economy/economic_system#1' , 2) } ,
    support = 0.14 , confidence = 0.83 , similarity = 0.0821
Rule 324:   {('share/portion#2' , 0) } -> {('country/state#1' , 1) } , support =
    0.17 , confidence = 0.50 , similarity = 0.0821
Rule 325:   {('company#1' 'country/state#1' , 0) } -> {('law/jurisprudence#2' , 1)
     } , support = 0.11 , confidence = 0.57 , similarity = 0.0829
Rule 326:   {('law/jurisprudence#2' , 1) ('share/portion#2' , 0) } -> {('economy/
    economic_system#1' , 2) } , support = 0.14 , confidence = 0.83 , similarity
     = 0.0841
Rule 327:   {('country/state#1' , 0) } -> {('commercial_enterprise/
    business_enterprise#2' , 1) } , support = 0.22 , confidence = 0.57 ,
    similarity = 0.0847
Rule 328:   {('plan/program#1' , 0) } -> {('country/state#1' , 1) } , support =
    0.11 , confidence = 0.50 , similarity = 0.0866
Rule 329:   {('time_period/period_of_time#1' , 0) } -> {('economy/economic_system
    #1' , 2) } , support = 0.11 , confidence = 0.50 , similarity = 0.0876
Rule 330:   {('datum/data_point#1' , 0) } -> {('country/state#1' , 1) } , support
     = 0.14 , confidence = 0.50 , similarity = 0.0883
Rule 331:   {('reform#1' , 0) } -> {('commercial_enterprise/business_enterprise#2'
     , 1) } , support = 0.14 , confidence = 0.56 , similarity = 0.0884
Rule 332:   {('time_period/period_of_time#1' , 0) } -> {('share/portion#2' , 2) }
    , support = 0.14 , confidence = 0.63 , similarity = 0.0924
Rule 333:   {('europe/nnp' 'company#1' , 0) } -> {('law/jurisprudence#2' , 1) } ,
     support = 0.11 , confidence = 0.57 , similarity = 0.0929
Rule 334:   {('yen#2' , 0) } -> {('euro#1' , 1) } , support = 0.11 , confidence
    = 0.57 , similarity = 0.0981
Rule 335:   {('law/jurisprudence#2' 'eu/nnp' , 0) } -> {('economy/economic_system
    #1' , 1) } , support = 0.17 , confidence = 0.75 , similarity = 0.0987
Rule 336:   {('law/jurisprudence#2' 'iraq/nnp' , 0) } -> {('economy/
    economic_system#1' , 1) } , support = 0.11 , confidence = 0.57 , similarity
     = 0.0987
Rule 337:   {('economy/economic_system#1' , 0) } -> {('law/jurisprudence#2' , 1) }
     , support = 0.19 , confidence = 0.50 , similarity = 0.0987
Rule 338:   {('iraq/nnp' , 0) ('economy/economic_system#1' , 1) } -> {('law/
    jurisprudence#2' , 2) } , support = 0.11 , confidence = 0.50 , similarity =
     0.0987
Rule 339:   {('occupation/business#1' , 0) } -> {('market/marketplace#1' , 1) ('
    economy/economic_system#1' , 2) } , support = 0.11 , confidence = 0.50 ,
    similarity = 0.1010
Rule 340:   {('iraq/nnp' , 1) ('company#1' , 0) } -> {('economy/economic_system#1'
     , 2) } , support = 0.11 , confidence = 0.57 , similarity = 0.1020
Rule 341:   {('iraq/nnp' , 0) ('economy/economic_system#1' , 1) } -> {('company#1'
     , 2) } , support = 0.11 , confidence = 0.50 , similarity = 0.1020
Rule 342:   {('state#4' , 0) } -> {('people#1' , 2) } , support = 0.11 ,
    confidence = 0.57 , similarity = 0.1172
Rule 343:   {('occupation/business#1' , 0) } -> {('market/marketplace#1' , 1) } ,
     support = 0.14 , confidence = 0.63 , similarity = 0.1275
Rule 344:   {('country/state#1' 'government/governing#3' , 0) } -> {('
    commercial_enterprise/business_enterprise#2' , 1) } , support = 0.11 ,
    confidence = 0.50 , similarity = 0.1390
Rule 345:   {('monetary_unit#1' , 0) } -> {('yen#2' , 1) } , support = 0.11 ,
    confidence = 0.57 , similarity = 0.1708
Rule 346:   {('euro#1' 'europe/nnp' , 0) } -> {('monetary_unit#1' , 1) } ,
    support = 0.11 , confidence = 0.57 , similarity = 0.2304
Rule 347:   {('time_period/period_of_time#1' , 0) } -> {('year#3' , 2) } ,
    support = 0.11 , confidence = 0.50 , similarity = 0.2917
```

C.3 Experiment 2: New FT

This section shows the results of experiments on the new Financial Times dataset.

```
********** TTM Testbench Results ************************************************

Dataset:
Name: New Financial Times
Time Granularity: days
Language: English
DataSet size: 2816

FITI Settings:
Minimum support: 0.1
Maximum support: 0.5
Minimum confidence: 0.5
Maximum confidence: 1.0
Maxspan: 3
Max set size: 3
Use only K first documents: -1
Prune rules by order of time: true

Rules: (56)
Rule 1:  {('europe/nnp' , 0) } -> {('market/marketplace#1' , 1) } , support =
    0.16 , confidence = 0.52 , similarity = 0.0000
Rule 2:  {('china/nnp' , 0) } -> {('military/armed_forces#1' , 1) } , support =
    0.21 , confidence = 0.54 , similarity = 0.0000
Rule 3:  {('russia/nnp' , 0) } -> {('military/armed_forces#1' , 1) } , support =
     0.12 , confidence = 0.54 , similarity = 0.0000
Rule 4:  {('iraq/nnp' , 0) } -> {('military/armed_forces#1' , 1) } , support =
    0.10 , confidence = 0.52 , similarity = 0.0000
Rule 5:  {('uk/nnp' , 0) } -> {('military/armed_forces#1' , 1) } , support =
    0.19 , confidence = 0.53 , similarity = 0.0000
Rule 6:  {('year#3' , 0) } -> {('china/nnp' , 1) } , support = 0.10 ,
    confidence = 0.50 , similarity = 0.0000
Rule 7:  {('europe/nnp' , 0) } -> {('china/nnp' , 1) } , support = 0.16 ,
    confidence = 0.52 , similarity = 0.0000
Rule 8:  {('year#3' , 0) } -> {('europe/nnp' , 1) } , support = 0.11 ,
    confidence = 0.55 , similarity = 0.0000
Rule 9:  {('commercial_enterprise/business_enterprise#2' , 0) } -> {('eu/nnp' ,
    1) } , support = 0.13 , confidence = 0.52 , similarity = 0.0000
Rule 10:  {('uk/nnp' , 0) } -> {('eu/nnp' , 1) } , support = 0.18 , confidence
    = 0.50 , similarity = 0.0000
Rule 11:  {('depository_financial_institution/bank#1' , 0) } -> {('eu/nnp' , 2) }
     , support = 0.13 , confidence = 0.52 , similarity = 0.0000
Rule 12:  {('russia/nnp' , 0) } -> {('military/armed_forces#1' , 2) } , support
    = 0.14 , confidence = 0.62 , similarity = 0.0000
Rule 13:  {('iraq/nnp' , 0) } -> {('military/armed_forces#1' , 2) } , support =
    0.11 , confidence = 0.57 , similarity = 0.0000
Rule 14:  {('sarkozy/nnp' , 0) } -> {('military/armed_forces#1' , 2) } , support
     = 0.13 , confidence = 0.56 , similarity = 0.0000
Rule 15:  {('uk/nnp' , 0) } -> {('military/armed_forces#1' , 2) } , support =
    0.21 , confidence = 0.58 , similarity = 0.0000
Rule 16:  {('europe/nnp' , 0) } -> {('military/armed_forces#1' , 1) } , support
    = 0.18 , confidence = 0.58 , similarity = 0.0000
Rule 17:  {('russia/nnp' , 0) } -> {('head/chief#4' , 2) } , support = 0.11 ,
    confidence = 0.50 , similarity = 0.0000
Rule 18:  {('russia/nnp' , 0) } -> {('president_of_the_united_states/
    united_states_president#1' , 2) } , support = 0.12 , confidence = 0.54 ,
    similarity = 0.0000
```

```
Rule 19:  {('russia/nnp' , 0) } -> {('head/chief#4' , 1) } ,  support = 0.11 ,
    confidence = 0.50 ,  similarity = 0.0000
Rule 20:  {('eu/nnp' , 0) } -> {('military/armed_forces#1' , 2) } ,  support =
    0.21 ,  confidence = 0.55 ,  similarity = 0.0000
Rule 21:  {('china/nnp' 'market/marketplace#1' , 0) } -> {('military/armed_forces
    #1' , 1) } ,  support = 0.11 ,  confidence = 0.55 ,  similarity = 0.0593
Rule 22:  {('market/marketplace#1' , 0) } -> {('military/armed_forces#1' , 2) } ,
      support = 0.23 ,  confidence = 0.59 ,  similarity = 0.0593
Rule 23:  {('market/marketplace#1' , 0) } -> {('military/armed_forces#1' , 1) } ,
      support = 0.23 ,  confidence = 0.59 ,  similarity = 0.0593
Rule 24:  {('china/nnp' 'market/marketplace#1' , 0) } -> {('military/armed_forces
    #1' , 2) } ,  support = 0.12 ,  confidence = 0.59 ,  similarity = 0.0593
Rule 25:  {('company#1' , 0) } -> {('market/marketplace#1' , 2) } ,  support =
    0.15 ,  confidence = 0.50 ,  similarity = 0.0600
Rule 26:  {('investor#1' , 1) ('uk/nnp' , 0) } -> {('military/armed_forces#1' ,
    2) } ,  support = 0.10 ,  confidence = 0.85 ,  similarity = 0.0600
Rule 27:  {('investor#1' 'uk/nnp' , 0) } -> {('military/armed_forces#1' , 1) } ,
     support = 0.12 ,  confidence = 0.72 ,  similarity = 0.0600
Rule 28:  {('investor#1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support
    = 0.28 ,  confidence = 0.69 ,  similarity = 0.0600
Rule 29:  {('investor#1' , 0) } -> {('military/armed_forces#1' , 2) } ,  support
    = 0.22 ,  confidence = 0.55 ,  similarity = 0.0600
Rule 30:  {('investor#1' 'uk/nnp' , 0) } -> {('military/armed_forces#1' , 2) } ,
     support = 0.11 ,  confidence = 0.67 ,  similarity = 0.0600
Rule 31:  {('week/hebdomad#1' , 0) } -> {('military/armed_forces#1' , 1) } ,
    support = 0.11 ,  confidence = 0.52 ,  similarity = 0.0607
Rule 32:  {('investor#1' 'president_of_the_united_states/united_states_president
    #1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support = 0.14 ,
    confidence = 0.83 ,  similarity = 0.0611
Rule 33:  {('china/nnp' , 0) ('president_of_the_united_states/
    united_states_president#1' , 1) } -> {('military/armed_forces#1' , 2) } ,
    support = 0.11 ,  confidence = 0.75 ,  similarity = 0.0622
Rule 34:  {('china/nnp' 'president_of_the_united_states/united_states_president
    #1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support = 0.11 ,
    confidence = 0.71 ,  similarity = 0.0622
Rule 35:  {('president_of_the_united_states/united_states_president#1' , 0) } ->
    {('military/armed_forces#1' , 1) } ,  support = 0.23 ,  confidence = 0.63 ,
    similarity = 0.0622
Rule 36:  {('investor#1' 'head/chief#4' , 0) } -> {('military/armed_forces#1' ,
    1) } ,  support = 0.12 ,  confidence = 0.72 ,  similarity = 0.0635
Rule 37:  {('conflict/struggle#1' , 0) } -> {('military/armed_forces#1' , 2) } ,
     support = 0.10 ,  confidence = 0.65 ,  similarity = 0.0637
Rule 38:  {('year#3' , 0) } -> {('military/armed_forces#1' , 1) } ,  support =
    0.11 ,  confidence = 0.55 ,  similarity = 0.0638
Rule 39:  {('head/chief#4' , 0) } -> {('military/armed_forces#1' , 1) } ,
    support = 0.20 ,  confidence = 0.64 ,  similarity = 0.0670
Rule 40:  {('head/chief#4' , 0) } -> {('military/armed_forces#1' , 2) } ,
    support = 0.17 ,  confidence = 0.55 ,  similarity = 0.0670
Rule 41:  {('commercial_enterprise/business_enterprise#2' , 0) } -> {('military/
    armed_forces#1' , 2) } ,  support = 0.18 ,  confidence = 0.70 ,  similarity =
    0.0674
Rule 42:  {('occupation/business#1' , 0) } -> {('military/armed_forces#1' , 2) }
    ,  support = 0.11 ,  confidence = 0.60 ,  similarity = 0.0703
Rule 43:  {('military/armed_forces#1' , 1) ('president_of_the_united_states/
    united_states_president#1' , 0) } -> {('investor#1' , 2) } ,  support = 0.14 ,
      confidence = 0.62 ,  similarity = 0.0735
Rule 44:  {('time#5' , 0) } -> {('military/armed_forces#1' , 1) } ,  support =
    0.11 ,  confidence = 0.60 ,  similarity = 0.0742
Rule 45:  {('time#5' , 0) } -> {('military/armed_forces#1' , 2) } ,  support =
    0.10 ,  confidence = 0.55 ,  similarity = 0.0742
```

```
Rule 46:  {('military/armed_forces#1' , 1) ('head/chief#4' , 0) } -> {('investor
    #1' , 2) } ,  support = 0.12 ,  confidence = 0.62 ,  similarity = 0.0784
Rule 47:  {('military/armed_forces#1' 'head/chief#4' , 0) } -> {('investor#1' ,
    2) } ,  support = 0.10 ,  confidence = 0.61 ,  similarity = 0.0784
Rule 48:  {('country/state#1' , 0) } -> {('investor#1' , 2) } ,  support = 0.12 ,
      confidence = 0.62 ,  similarity = 0.0817
Rule 49:  {('president_of_the_united_states/united_states_president#1' , 0) } ->
    {('investor#1' , 2) } ,  support = 0.19 ,  confidence = 0.53 ,  similarity =
    0.0870
Rule 50:  {('company#1' 'president_of_the_united_states/united_states_president
    #1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support = 0.13 ,
    confidence = 0.74 ,  similarity = 0.0873
Rule 51:  {('company#1' 'president_of_the_united_states/united_states_president
    #1' , 0) } -> {('military/armed_forces#1' , 2) } ,  support = 0.10 ,
    confidence = 0.58 ,  similarity = 0.0873
Rule 52:  {('president_of_the_united_states/united_states_president#1' 'head/
    chief#4' , 0) } -> {('investor#1' , 2) } ,  support = 0.11 ,  confidence =
    0.71 ,  similarity = 0.0919
Rule 53:  {('head/chief#4' , 0) } -> {('investor#1' , 2) } ,  support = 0.17 ,
    confidence = 0.55 ,  similarity = 0.0968
Rule 54:  {('depository_financial_institution/bank#1' , 0) } -> {('military/
    armed_forces#1' , 2) } ,  support = 0.13 ,  confidence = 0.52 ,  similarity =
    0.0973
Rule 55:  {('company#1' , 0) } -> {('military/armed_forces#1' , 2) } ,  support =
     0.16 ,  confidence = 0.53 ,  similarity = 0.1124
Rule 56:  {('company#1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support =
     0.17 ,  confidence = 0.56 ,  similarity = 0.1124
```

C.4 Experiment 3: BBC

This section shows the results of experiments on the BBC Business dataset.

```
********** TTM Testbench Results ********************************************

Dataset:
Name: BBC Finance News
Time Granularity: days
Language: English
DataSet size: 1528

FITI Settings:
Minimum support: 0.1
Maximum support: 0.5
Minimum confidence: 0.5
Maximum confidence: 1.0
Maxspan: 3
Max set size: 3
Use only K first documents: -1
Prune rules by order of time: true

Rules: (21)
Rule 1:  {('net_income/net#1' , 0) } -> {('uk/nnp' , 2) } ,  support = 0.19 ,
    confidence = 0.54 ,  similarity = 0.0000
Rule 2:  {('eu/nnp' , 0) } -> {('uk/nnp' , 1) } ,  support = 0.16 ,  confidence =
     0.50 ,  similarity = 0.0000
Rule 3:  {('net_income/net#1' , 0) } -> {('uk/nnp' , 1) } ,  support = 0.18 ,
    confidence = 0.50 ,  similarity = 0.0000
```

```
Rule 4:   {('rate#2' , 0) } -> {('uk/nnp' , 2) } ,  support = 0.10 ,  confidence =
     0.50 ,  similarity = 0.0000
Rule 5:   {('rate#2' , 0) } -> {('eu/nnp' , 1) } ,  support = 0.13 ,  confidence =
     0.62 ,  similarity = 0.0000
Rule 6:   {('rate#2' , 0) } -> {('uk/nnp' , 1) } ,  support = 0.10 ,  confidence =
     0.50 ,  similarity = 0.0000
Rule 7:   {('japan/nnp' , 0) } -> {('firm/house#1' , 1) } ,  support = 0.13 ,
    confidence = 0.53 ,  similarity = 0.0000
Rule 8:   {('share/portion#2' , 0) } -> {('uk/nnp' , 1) } ,  support = 0.12 ,
    confidence = 0.53 ,  similarity = 0.0000
Rule 9:   {('japan/nnp' 'uk/nnp' , 0) } -> {('firm/house#1' , 2) } ,  support =
    0.10 ,  confidence = 0.62 ,  similarity = 0.0000
Rule 10:   {('eu/nnp' , 0) } -> {('firm/house#1' , 2) } ,  support = 0.16 ,
    confidence = 0.50 ,  similarity = 0.0000
Rule 11:   {('firm/house#1' , 1) ('net_income/net#1' , 0) } -> {('uk/nnp' , 2) } ,
      support = 0.10 ,  confidence = 0.67 ,  similarity = 0.0000
Rule 12:   {('china/nnp' , 0) } -> {('uk/nnp' , 2) } ,  support = 0.21 ,
    confidence = 0.57 ,  similarity = 0.0000
Rule 13:   {('china/nnp' , 0) } -> {('firm/house#1' , 1) } ,  support = 0.19 ,
    confidence = 0.54 ,  similarity = 0.0000
Rule 14:   {('drop/dip#3' , 0) } -> {('firm/house#1' , 1) } ,  support = 0.10 ,
    confidence = 0.57 ,  similarity = 0.0572
Rule 15:   {('uk/nnp' 'net_income/net#1' , 0) } -> {('sale/cut-rate_sale#4' , 1) }
     ,  support = 0.12 ,  confidence = 0.50 ,  similarity = 0.0605
Rule 16:   {('uk/nnp' , 0) ('net_income/net#1' , 1) } -> {('sale/cut-rate_sale#4'
    , 2) } ,  support = 0.12 ,  confidence = 0.53 ,  similarity = 0.0605
Rule 17:   {('rate#2' , 0) } -> {('net_income/net#1' , 1) } ,  support = 0.10 ,
    confidence = 0.50 ,  similarity = 0.0664
Rule 18:   {('uk/nnp' , 1) ('net_income/net#1' , 0) } -> {('firm/house#1' , 2) } ,
      support = 0.10 ,  confidence = 0.57 ,  similarity = 0.0683
Rule 19:   {('firm/house#1' , 1) ('uk/nnp' , 0) } -> {('net_income/net#1' , 2) } ,
      support = 0.10 ,  confidence = 0.53 ,  similarity = 0.0683
Rule 20:   {('occupation/business#1' , 0) } -> {('net_income/net#1' , 1) } ,
    support = 0.12 ,  confidence = 0.50 ,  similarity = 0.0745
Rule 21:   {('depository_financial_institution/bank#1' , 0) } -> {('firm/house#1'
    , 1) } ,  support = 0.16 ,  confidence = 0.75 ,  similarity = 0.1099
```

C.5 Experiment 4: New FT With Adjusted FITI Parameters

This section shows the results of experiments on the new Financial Times dataset, with lower support and higher confidence than experiment 2.

```
********** TTM Testbench Results ********************************************

Dataset:
Name: New Financial Times
Time Granularity: days
Language: English
DataSet size: 2816

FITI Settings:
Minimum support: 0.065
Maximum support: 0.5
Minimum confidence: 0.7
Maximum confidence: 1.0
Maxspan: 3
Max set size: 3
```

```
Use only K first documents: -1
Prune rules by order of time: true


Rules: (71)
Rule 1:   {('america/nnp' , 0) } -> {('market/marketplace#1' , 1) } ,  support =
    0.08 ,  confidence = 0.73 ,  similarity = 0.0000
Rule 2:   {('ti/nnp' , 0) } -> {('military/armed_forces#1' , 2) } ,  support =
    0.07 ,  confidence = 0.78 ,  similarity = 0.0000
Rule 3:   {('uk/nnp' 'iraq/nnp' , 0) } -> {('military/armed_forces#1' , 2) } ,
    support = 0.07 ,  confidence = 0.70 ,  similarity = 0.0000
Rule 4:   {('president/chairman#4' , 0) } -> {('uk/nnp' , 2) } ,  support = 0.10 ,
      confidence = 0.71 ,  similarity = 0.0000
Rule 5:   {('europe/nnp' 'uk/nnp' , 0) } -> {('military/armed_forces#1' , 1) } ,
    support = 0.07 ,  confidence = 0.70 ,  similarity = 0.0000
Rule 6:   {('carrefour/nnp' , 0) } -> {('military/armed_forces#1' , 1) } ,
    support = 0.07 ,  confidence = 0.78 ,  similarity = 0.0000
Rule 7:   {('abn/nnp' , 0) } -> {('military/armed_forces#1' , 1) } ,  support =
    0.10 ,  confidence = 0.79 ,  similarity = 0.0000
Rule 8:   {('military/armed_forces#1' , 1) ('election#1' , 0) } -> {('uk/nnp' , 2)
     } ,  support = 0.07 ,  confidence = 0.70 ,  similarity = 0.0000
Rule 9:   {('europe/nnp' , 1) ('iraq/nnp' , 0) } -> {('military/armed_forces#1' ,
    2) } ,  support = 0.07 ,  confidence = 0.78 ,  similarity = 0.0000
Rule 10:   {('europe/nnp' , 1) ('uk/nnp' , 0) } -> {('military/armed_forces#1' ,
    2) } ,  support = 0.10 ,  confidence = 0.71 ,  similarity = 0.0000
Rule 11:   {('attempt/effort#1' , 0) } -> {('china/nnp' , 2) } ,  support = 0.07 ,
      confidence = 0.70 ,  similarity = 0.0000
Rule 12:   {('china/nnp' 'year#3' , 0) } -> {('eu/nnp' , 2) } ,  support = 0.09 ,
     confidence = 0.75 ,  similarity = 0.0000
Rule 13:   {('china/nnp' 'biology/biological_science#1' , 0) } -> {('eu/nnp' , 2)
    } ,  support = 0.07 ,  confidence = 0.87 ,  similarity = 0.0000
Rule 14:   {('year#3' , 0) } -> {('eu/nnp' , 2) } ,  support = 0.15 ,  confidence
    = 0.73 ,  similarity = 0.0000
Rule 15:   {('hedge_fund/hedgefund#1' , 0) } -> {('military/armed_forces#1' , 2) }
     ,  support = 0.10 ,  confidence = 0.77 ,  similarity = 0.0539
Rule 16:   {('stock_exchange/stock_market#1' , 0) } -> {('military/armed_forces#1'
     , 2) } ,  support = 0.08 ,  confidence = 0.80 ,  similarity = 0.0543
Rule 17:   {('crisis#1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support =
    0.10 ,  confidence = 0.71 ,  similarity = 0.0558
Rule 18:   {('election#1' , 0) } -> {('investor#1' , 2) } ,  support = 0.11 ,
    confidence = 0.80 ,  similarity = 0.0588
Rule 19:   {('china/nnp' 'election#1' , 0) } -> {('investor#1' , 2) } ,  support =
     0.07 ,  confidence = 0.87 ,  similarity = 0.0588
Rule 20:   {('market/marketplace#1' 'election#1' , 0) } -> {('military/
    armed_forces#1' , 1) } ,  support = 0.07 ,  confidence = 0.87 ,  similarity =
    0.0591
Rule 21:   {('eu/nnp' 'market/marketplace#1' , 0) } -> {('military/armed_forces#1'
     , 2) } ,  support = 0.10 ,  confidence = 0.79 ,  similarity = 0.0593
Rule 22:   {('china/nnp' , 1) ('market/marketplace#1' , 0) } -> {('military/
    armed_forces#1' , 2) } ,  support = 0.08 ,  confidence = 0.73 ,  similarity =
    0.0593
Rule 23:   {('investor#1' , 0) ('market/marketplace#1' , 1) } -> {('military/
    armed_forces#1' , 2) } ,  support = 0.11 ,  confidence = 0.75 ,  similarity =
    0.0597
Rule 24:   {('investor#1' , 1) ('market/marketplace#1' , 0) } -> {('military/
    armed_forces#1' , 2) } ,  support = 0.14 ,  confidence = 0.83 ,  similarity =
    0.0597
Rule 25:   {('investor#1' 'market/marketplace#1' , 0) } -> {('military/
    armed_forces#1' , 1) } ,  support = 0.13 ,  confidence = 0.82 ,  similarity =
    0.0597
```

```
Rule 26:  {('investor#1' , 1) ('russia/nnp' , 0) } -> {('military/armed_forces#1'
    , 2) } , support = 0.07 , confidence = 0.78 , similarity = 0.0600
Rule 27:  {('investor#1' , 1) ('uk/nnp' , 0) } -> {('military/armed_forces#1' ,
   2) } , support = 0.10 , confidence = 0.85 , similarity = 0.0600
Rule 28:  {('eu/nnp' , 0) ('investor#1' , 1) } -> {('military/armed_forces#1' ,
   2) } , support = 0.09 , confidence = 0.90 , similarity = 0.0600
Rule 29:  {('eu/nnp' 'investor#1' , 0) } -> {('military/armed_forces#1' , 2) } ,
    support = 0.12 , confidence = 0.72 , similarity = 0.0600
Rule 30:  {('investor#1' 'sarkozy/nnp' , 0) } -> {('military/armed_forces#1' , 2)
    } , support = 0.09 , confidence = 0.75 , similarity = 0.0600
Rule 31:  {('investor#1' 'russia/nnp' , 0) } -> {('military/armed_forces#1' , 2)
   } , support = 0.07 , confidence = 0.87 , similarity = 0.0600
Rule 32:  {('china/nnp' , 0) ('investor#1' , 1) } -> {('military/armed_forces#1'
   , 2) } , support = 0.10 , confidence = 0.85 , similarity = 0.0600
Rule 33:  {('investor#1' 'uk/nnp' , 0) } -> {('military/armed_forces#1' , 1) } ,
    support = 0.12 , confidence = 0.72 , similarity = 0.0600
Rule 34:  {('investor#1' 'iraq/nnp' , 0) } -> {('military/armed_forces#1' , 1) }
    , support = 0.10 , confidence = 0.77 , similarity = 0.0600
Rule 35:  {('investor#1' 'abn/nnp' , 0) } -> {('military/armed_forces#1' , 1) } ,
     support = 0.07 , confidence = 0.78 , similarity = 0.0600
Rule 36:  {('china/nnp' 'investor#1' , 0) } -> {('military/armed_forces#1' , 1) }
    , support = 0.11 , confidence = 0.75 , similarity = 0.0600
Rule 37:  {('market/marketplace#1' , 0) ('president_of_the_united_states/
   united_states_president#1' , 1) } -> {('military/armed_forces#1' , 2) } ,
   support = 0.10 , confidence = 0.73 , similarity = 0.0607
Rule 38:  {('market/marketplace#1' 'president_of_the_united_states/
   united_states_president#1' , 0) } -> {('military/armed_forces#1' , 1) } ,
   support = 0.10 , confidence = 0.71 , similarity = 0.0607
Rule 39:  {('eu/nnp' , 0) ('talk/talking#1' , 1) } -> {('investor#1' , 2) } ,
   support = 0.07 , confidence = 0.78 , similarity = 0.0608
Rule 40:  {('investor#1' 'president_of_the_united_states/united_states_president
   #1' , 0) } -> {('military/armed_forces#1' , 1) } , support = 0.14 ,
   confidence = 0.83 , similarity = 0.0611
Rule 41:  {('investor#1' 'conflict/struggle#1' , 0) } -> {('military/armed_forces
   #1' , 1) } , support = 0.07 , confidence = 0.78 , similarity = 0.0619
Rule 42:  {('market/marketplace#1' 'day#4' , 0) } -> {('military/armed_forces#1'
   , 2) } , support = 0.07 , confidence = 0.70 , similarity = 0.0621
Rule 43:  {('china/nnp' , 0) ('president_of_the_united_states/
   united_states_president#1' , 1) } -> {('military/armed_forces#1' , 2) } ,
   support = 0.11 , confidence = 0.75 , similarity = 0.0622
Rule 44:  {('uk/nnp' 'president_of_the_united_states/united_states_president#1' ,
    0) } -> {('military/armed_forces#1' , 1) } , support = 0.10 , confidence =
   0.71 , similarity = 0.0622
Rule 45:  {('china/nnp' 'president_of_the_united_states/united_states_president
   #1' , 0) } -> {('military/armed_forces#1' , 1) } , support = 0.11 ,
   confidence = 0.71 , similarity = 0.0622
Rule 46:  {('commercial_enterprise/business_enterprise#2' 'market/marketplace#1'
   , 0) } -> {('military/armed_forces#1' , 2) } , support = 0.10 , confidence =
    0.77 , similarity = 0.0634
Rule 47:  {('commercial_enterprise/business_enterprise#2' , 0) ('market/
   marketplace#1' , 1) } -> {('military/armed_forces#1' , 2) } , support = 0.10
   , confidence = 1.00 , similarity = 0.0634
Rule 48:  {('investor#1' , 0) ('head/chief#4' , 1) } -> {('military/armed_forces
   #1' , 2) } , support = 0.10 , confidence = 0.77 , similarity = 0.0635
Rule 49:  {('investor#1' 'head/chief#4' , 0) } -> {('military/armed_forces#1' ,
   1) } , support = 0.12 , confidence = 0.72 , similarity = 0.0635
Rule 50:  {('president_of_the_united_states/united_states_president#1' 'head/
   chief#4' , 0) } -> {('military/armed_forces#1' , 1) } , support = 0.11 ,
   confidence = 0.71 , similarity = 0.0646
Rule 51:  {('uk/nnp' , 0) ('head/chief#4' , 1) } -> {('military/armed_forces#1' ,
    2) } , support = 0.10 , confidence = 0.77 , similarity = 0.0670
```

```
Rule 52:   {('russia/nnp' , 0) ('head/chief#4' , 1) } -> {('military/armed_forces
    #1' , 2) } ,  support = 0.10 ,  confidence = 0.83 ,  similarity = 0.0670
Rule 53:   {('russia/nnp' 'head/chief#4' , 0) } -> {('military/armed_forces#1' ,
    1) } ,  support = 0.08 ,  confidence = 0.73 ,  similarity = 0.0670
Rule 54:   {('uk/nnp' 'head/chief#4' , 0) } -> {('military/armed_forces#1' , 1) }
    ,  support = 0.10 ,  confidence = 0.79 ,  similarity = 0.0670
Rule 55:   {('eu/nnp' 'commercial_enterprise/business_enterprise#2' , 0) } -> {('
    military/armed_forces#1' , 2) } ,  support = 0.10 ,  confidence = 0.85 ,
    similarity = 0.0674
Rule 56:   {('eu/nnp' , 1) ('commercial_enterprise/business_enterprise#2' , 0) }
    -> {('military/armed_forces#1' , 2) } ,  support = 0.10 ,  confidence = 0.71 ,
     similarity = 0.0674
Rule 57:   {('commercial_enterprise/business_enterprise#2' , 0) } -> {('military/
    armed_forces#1' , 2) } ,  support = 0.18 ,  confidence = 0.70 ,  similarity =
    0.0674
Rule 58:   {('president_of_the_united_states/united_states_president#1' 'country/
    state#1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support = 0.07 ,
    confidence = 0.70 ,  similarity = 0.0679
Rule 59:   {('military/armed_forces#1' , 1) ('election#1' , 0) } -> {('market/
    marketplace#1' , 2) } ,  support = 0.07 ,  confidence = 0.70 ,  similarity =
    0.0704
Rule 60:   {('head/chief#4' , 0) ('time#5' , 1) } -> {('military/armed_forces#1' ,
     2) } ,  support = 0.07 ,  confidence = 0.70 ,  similarity = 0.0706
Rule 61:   {('power/powerfulness#1' , 0) } -> {('head/chief#4' , 2) } ,  support =
     0.09 ,  confidence = 0.75 ,  similarity = 0.0727
Rule 62:   {('market/marketplace#1' 'president_of_the_united_states/
    united_states_president#1' , 0) } -> {('investor#1' , 2) } ,  support = 0.10 ,
     confidence = 0.71 ,  similarity = 0.0731
Rule 63:   {('company#1' 'country/state#1' , 0) } -> {('
    president_of_the_united_states/united_states_president#1' , 2) } ,  support =
    0.07 ,  confidence = 0.70 ,  similarity = 0.0744
Rule 64:   {('company#1' 'head/chief#4' , 0) } -> {('
    president_of_the_united_states/united_states_president#1' , 2) } ,  support =
    0.08 ,  confidence = 0.73 ,  similarity = 0.0828
Rule 65:   {('president_of_the_united_states/united_states_president#1' 'country/
    state#1' , 0) } -> {('investor#1' , 2) } ,  support = 0.07 ,  confidence =
    0.70 ,  similarity = 0.0844
Rule 66:   {('investor#1' 'company#1' , 0) } -> {('military/armed_forces#1' , 1) }
     ,  support = 0.12 ,  confidence = 0.81 ,  similarity = 0.0862
Rule 67:   {('company#1' 'president_of_the_united_states/united_states_president
    #1' , 0) } -> {('military/armed_forces#1' , 1) } ,  support = 0.13 ,
    confidence = 0.74 ,  similarity = 0.0873
Rule 68:   {('president_of_the_united_states/united_states_president#1' 'head/
    chief#4' , 0) } -> {('investor#1' , 2) } ,  support = 0.11 ,  confidence =
    0.71 ,  similarity = 0.0919
Rule 69:   {('uk/nnp' 'head/chief#4' , 0) } -> {('investor#1' , 2) } ,  support =
    0.10 ,  confidence = 0.71 ,  similarity = 0.0968
Rule 70:   {('uk/nnp' , 1) ('head/chief#4' , 0) } -> {('investor#1' , 2) } ,
    support = 0.08 ,  confidence = 0.73 ,  similarity = 0.0968
Rule 71:   {('group/grouping#1' , 0) } -> {('military/armed_forces#1' , 2) } ,
    support = 0.12 ,  confidence = 0.72 ,  similarity = 0.1848
```

C.6 Experiment 5: BBC With Adjusted FITI Parameters

This section shows the results of experiments on the BBC Business dataset, with lower support and higher confidence than experiment 3.

```
********** TTM Testbench Results *******************************************

Dataset:
Name: BBC Finance News
Time Granularity: days
Language: English
DataSet size: 1528

FITI Settings:
Minimum support: 0.06
Maximum support: 0.5
Minimum confidence: 0.7
Maximum confidence: 1.0
Maxspan: 3
Max set size: 3
Use only K first documents: -1
Prune rules by order of time: true


Rules: (43)
Rule 1:   {('rate#2' 'uk/nnp' , 0) } -> {('eu/nnp' , 1) } , support = 0.10 ,
    confidence = 0.73 , similarity = 0.0000
Rule 2:   {('rate#2' 'sale/cut-rate_sale#4' , 0) } -> {('eu/nnp' , 1) } , support
     = 0.09 , confidence = 0.78 , similarity = 0.0000
Rule 3:   {('india/nnp' , 1) ('uk/nnp' , 0) } -> {('firm/house#1' , 2) } ,
    support = 0.06 , confidence = 1.00 , similarity = 0.0000
Rule 4:   {('uk/nnp' 'robert_peston/nnp' , 0) } -> {('net_income/net#1' , 1) } ,
    support = 0.06 , confidence = 0.71 , similarity = 0.0000
Rule 5:   {('barclays/nnp' , 0) } -> {('net_income/net#1' , 1) } , support = 0.09
     , confidence = 0.78 , similarity = 0.0000
Rule 6:   {('wolfowitz/nnp' , 0) } -> {('net_income/net#1' , 1) } , support =
    0.08 , confidence = 0.86 , similarity = 0.0000
Rule 7:   {('virgin/nnp' , 0) } -> {('law/jurisprudence#2' , 1) } , support =
    0.06 , confidence = 0.71 , similarity = 0.0000
Rule 8:   {('country/state#1' , 0) } -> {('uk/nnp' , 1) } , support = 0.08 ,
    confidence = 0.75 , similarity = 0.0000
Rule 9:   {('country/state#1' , 0) } -> {('uk/nnp' , 2) } , support = 0.09 ,
    confidence = 0.88 , similarity = 0.0000
Rule 10:   {('worker#1' , 0) } -> {('uk/nnp' , 2) } , support = 0.09 ,
    confidence = 0.70 , similarity = 0.0000
Rule 11:   {('world_bank/nnp' , 0) } -> {('uk/nnp' , 2) } , support = 0.06 ,
    confidence = 0.71 , similarity = 0.0000
Rule 12:   {('rate#2' , 1) ('uk/nnp' , 0) } -> {('eu/nnp' , 2) } , support = 0.09
     , confidence = 1.00 , similarity = 0.0000
Rule 13:   {('rate#2' , 1) ('net_income/net#1' , 0) } -> {('uk/nnp' , 2) } ,
    support = 0.08 , confidence = 0.75 , similarity = 0.0000
Rule 14:   {('barclays/nnp' , 0) } -> {('net_income/net#1' , 2) } , support =
    0.09 , confidence = 0.78 , similarity = 0.0000
Rule 15:   {('wolfowitz/nnp' , 0) } -> {('net_income/net#1' , 2) } , support =
    0.06 , confidence = 0.71 , similarity = 0.0000
Rule 16:   {('abn/nnp' 'abn_amro/nnp' , 0) } -> {('net_income/net#1' , 2) } ,
    support = 0.06 , confidence = 0.71 , similarity = 0.0000
Rule 17:   {('apple/nnp' , 0) } -> {('firm/house#1' , 1) } , support = 0.06 ,
    confidence = 0.71 , similarity = 0.0000
Rule 18:   {('center/centre#1' , 0) } -> {('uk/nnp' , 1) } , support = 0.08 ,
    confidence = 0.75 , similarity = 0.0000
Rule 19:   {('world_bank/nnp' , 0) } -> {('uk/nnp' , 1) } , support = 0.08 ,
    confidence = 0.86 , similarity = 0.0000
Rule 20:   {('firm/house#1' 'interest_rate/rate_of_interest#1' , 0) } -> {('uk/nnp
    ' , 1) } , support = 0.06 , confidence = 0.83 , similarity = 0.0000
```

```
Rule 21:  {('worker#1' , 0) } -> {('uk/nnp' , 1) } , support = 0.10 ,
    confidence = 0.80 , similarity = 0.0000
Rule 22:  {('google/nnp' , 0) } -> {('uk/nnp' , 1) } , support = 0.09 ,
    confidence = 0.70 , similarity = 0.0000
Rule 23:  {('firm/house#1' 'robert_peston/nnp' , 0) } -> {('uk/nnp' , 1) } ,
    support = 0.06 , confidence = 0.71 , similarity = 0.0000
Rule 24:  {('india/nnp' 'firm/house#1' , 0) } -> {('uk/nnp' , 1) } , support =
    0.06 , confidence = 0.71 , similarity = 0.0000
Rule 25:  {('takeover_bid#1' , 0) } -> {('net_income/net#1' , 2) } , support =
    0.10 , confidence = 0.80 , similarity = 0.0493
Rule 26:  {('uk/nnp' , 0) ('stock_exchange/stock_market#1' , 1) } -> {('firm/
    house#1' , 2) } , support = 0.06 , confidence = 0.71 , similarity = 0.0581
Rule 27:  {('sale/cut-rate_sale#4' 'barclays/nnp' , 0) } -> {('net_income/net#1'
    , 1) } , support = 0.06 , confidence = 1.00 , similarity = 0.0605
Rule 28:  {('sale/cut-rate_sale#4' 'barclays/nnp' , 0) } -> {('net_income/net#1'
    , 2) } , support = 0.06 , confidence = 1.00 , similarity = 0.0605
Rule 29:  {('sale/cut-rate_sale#4' , 1) ('china/nnp' , 0) } -> {('firm/house#1' ,
     2) } , support = 0.08 , confidence = 0.75 , similarity = 0.0618
Rule 30:  {('sale/cut-rate_sale#4' 'china/nnp' , 0) } -> {('firm/house#1' , 1) }
    , support = 0.08 , confidence = 0.75 , similarity = 0.0618
Rule 31:  {('japan/nnp' 'sale/cut-rate_sale#4' , 0) } -> {('firm/house#1' , 1) }
    , support = 0.06 , confidence = 0.71 , similarity = 0.0618
Rule 32:  {('rate#2' , 1) ('uk/nnp' , 0) } -> {('net_income/net#1' , 2) } ,
    support = 0.06 , confidence = 0.71 , similarity = 0.0664
Rule 33:  {('occupation/business#1' 'uk/nnp' , 0) } -> {('net_income/net#1' , 1)
    } , support = 0.09 , confidence = 0.78 , similarity = 0.0745
Rule 34:  {('country/state#1' , 0) } -> {('net_income/net#1' , 2) } , support =
    0.08 , confidence = 0.75 , similarity = 0.0782
Rule 35:  {('depository_financial_institution/bank#1' , 1) ('sale/cut-rate_sale
    #4' , 0) } -> {('firm/house#1' , 2) } , support = 0.06 , confidence = 0.71 ,
      similarity = 0.0859
Rule 36:  {('depository_financial_institution/bank#1' 'sale/cut-rate_sale#4' , 0)
     } -> {('firm/house#1' , 1) } , support = 0.09 , confidence = 1.00 ,
    similarity = 0.0859
Rule 37:  {('depository_financial_institution/bank#1' 'occupation/business#1' ,
    0) } -> {('firm/house#1' , 1) } , support = 0.06 , confidence = 1.00 ,
    similarity = 0.0933
Rule 38:  {('depository_financial_institution/bank#1' , 1) ('china/nnp' , 0) } ->
     {('firm/house#1' , 2) } , support = 0.09 , confidence = 0.70 , similarity
    = 0.1099
Rule 39:  {('depository_financial_institution/bank#1' , 1) ('uk/nnp' , 0) } ->
    {('firm/house#1' , 2) } , support = 0.06 , confidence = 1.00 , similarity =
     0.1099
Rule 40:  {('depository_financial_institution/bank#1' , 0) } -> {('firm/house#1'
    , 1) } , support = 0.16 , confidence = 0.75 , similarity = 0.1099
Rule 41:  {('depository_financial_institution/bank#1' 'china/nnp' , 0) } -> {('
    firm/house#1' , 1) } , support = 0.12 , confidence = 0.90 , similarity =
    0.1099
Rule 42:  {('depository_financial_institution/bank#1' 'uk/nnp' , 0) } -> {('firm/
    house#1' , 1) } , support = 0.08 , confidence = 0.86 , similarity = 0.1099
Rule 43:  {('depository_financial_institution/bank#1' 'japan/nnp' , 0) } -> {('
    firm/house#1' , 1) } , support = 0.08 , confidence = 0.75 , similarity =
    0.1099
```

Bibliography

[1] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, pages 487–499, 1994.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] S. Banerjee and T. Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003.

[4] S. Basu et al. Evaluating the Novelty of TextMined Rules Using Lexical Knowledge. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 233–238, 2001.

[5] R. K. Bist, H. S. Dhami, and N. Tiwari. An Evaluation of Different Statistical Techniques of Collocation Extraction Using a Probability Measure to Word Combinations. *Journal of Quantitative Linguistics*, 13(2):161–175, 2006.

[6] P. Borst. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, University of Twente, 1997.

[7] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[8] A. Duineveld, R. Stoter, M. Weiden, B. Kenepa, and V. R. Benjamins. WonderTools? A Comparative Study of Ontological Engineering Tools. *International Journal of Human-Computer Studies*, 52(6):1111–1133, 2000.

[9] M. H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall, 2003.

[10] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17:37–54, 1996.

[11] R. Feldman and J. Sanger. *The Text Mining Handbook*. Cambridge, 2007.

[12] R. Feldman et al. Knowledge Management: A Text Mining Approach. In *Proceedings of the 2nd Conference on Practical Aspects of Knowledge Management*, 1998.

[13] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[14] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

[15] E. Hovy et al. OntoNotes: The 90% Solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 57–60, 2006.

[16] N. Ide and J. Veronis. Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1–40, 1998.

[17] J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. 1997.

[18] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* Prentice Hall, 2000.

[19] J. S. Justeson and S. Katz. Technical Terminology: Some Linguistics Properties and an Algorithm for Identification in Text. *Natural Language Engineering*, 1:9–27, 1995.

[20] H. Karanikas and B. Theodoulidis. Knowledge Discovery in Text and Text Mining Software. Technical report, Centre for Research in Information Management, Department of Computation, UMIST, 2002.

[21] Y. Kodratoff. Rating the Interest of Rules Induced From Data and Within Texts. In *Proceedings of the 12th International Workshop on Database and Expert Systems Applications*, pages 265–269, 2001.

[22] S. Laxman and P. S. Sastry. A Survey of Temporal Data Mining. *Sādhanā*, 31:173–198, 2006.

[23] M. Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *SIGDOC '86: Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26, 1986.

[24] C.-Y. Lin. Knowledge-Based Automatic Topic Identification. In *Meeting of the Association for Computational Linguistics*, pages 308–310, 1995.

[25] D. Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304, 1998.

[26] H. Lu, L. Feng, and J. Han. Beyond Intratransaction Association Analysis: Mining Multidimensional Intertransaction Association Rules. *ACM Trans. Inf. Syst.*, 18(4):423–454, 2000.

[27] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[28] Q. Mei and C. Zhai. Discovering Evolutionary Theme Patterns from Text: An Exploration of Temporal Text Mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 198–207, 2005.

[29] G. A. Miller et al. Introduction to WordNet: An On-line Lexical Database. Technical report, Cognitive Science Laboratory, Princeton University, 1993.

[30] R. J. Mooney and R. Bunescu. Mining Knowledge from Text Using Information Extraction. *SIGKDD Explor. Newsl.*, 7(1):3–10, 2005.

[31] R. J. Mooney and U. Y. Nahm. Text Mining with Information Extraction. In *Proceedings of the AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, 2002.

[32] K. Nørvåg. Supporting Temporal Text-Containment Queries in Temporal Document Databases. *Data Knowledge Engineering*, 49:105–125.

[33] K. Nørvåg. V2: A Database Approach to Temporal Document Management. In *Proceedings of the Seventh International Conference on Database Engineering and Applications Symposium*, pages 212–221, 2003.

[34] K. Nørvåg, K.-I. Skogstad, and T. Eriksen. Mining Association Rules in Temporal Document Collections. In *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems (ISMIS'2006)*, 2006.

[35] K. Nørvåg and R. Øyri. News Item Extraction for Text Mining in Web Newspapers. In *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration*, pages 195–204, 2005.

[36] F. Patman and P. Thompson. Names: A New Frontier in Text Mining. In *ISI*, pages 27–38, 2003.

[37] S. Patwardhan, S. Banerjee, and T. Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, 2003.

[38] Princeton University Cognitive Science Laboratory. WordNet [online]. Available from: `http://wordnet.princeton.edu/`.

[39] M. Pucher. Performance Evaluation of WordNet-based Semantic Relatedness Measures for Word Prediction in Conversational Speech. In *Sixth International Workshop on Computational Semantics*, 2005.

[40] J. Roddick and M. Spiliopoulou. A Survey of Temporal Knowledge Discovery Paradigms and Methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.

[41] R. Srikant and R. Agrawal. Mining Generalized Association Rules. *Future Generation Computer Systems*, 13(2–3):161–180, 1995.

[42] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.* Addison Wesley, 2006.

[43] K. Toutanova and C. Manning. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000.

[44] A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient Mining of Intertransaction Association Rules. *IEEE Transactions on Knowledge and Data Engineering*, pages 43–56, 2003.

[45] A. Voutilainen. A Syntax-based Part-of-Speech Analyser. In *Proceedings of the seventh Conference on European Chapter of the Association for Computational Linguistics*, pages 157–164, 1995.