Integration of Search Theories and Evidential Analysis to Web-wide Discovery of Information for Decision Support

by

Natalia Danilova, MSc (Eng) London

Systems and Control Engineering Centre

School of Mathematics, Computer Science and Engineering

City University London

November 2014

Declaration

The author grants powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Abstract

The main contribution of this research is that it addresses the issues associated with traditional information gathering and presents a novel semantic approach method to Web-based discovery of previously unknown intelligence for effective decision making. It provides a comprehensive theoretical background to the proposed solution together with a demonstration of the effectiveness of the method from results of the experiments, showing how the quality of collected information can be significantly enhanced by previously unknown information derived from the available known facts.

The quality of decisions made in business and government relates directly to the quality of the information used to formulate the decision. This information may be retrieved from an organisation's knowledge base (Intranet) or from the World Wide Web. The purpose of this thesis is to investigate the specifics of information gathering from these sources. It has studied a number of search techniques that rely on statistical and semantic analysis of unstructured information, and identified benefits and limitations of these techniques. It was concluded that enterprise search technologies can efficiently manipulate Intranet held information, but require complex processing of large amount of textual information, which is not feasible and scalable when applied to the Web.

Based upon the search methods investigations, this thesis introduces a new semantic Web-based search method that automates the correlation of topic-related content for discovery of hitherto unknown information from disparate and widely diverse Web-sources. This method is in contrast to traditional search methods that are constrained to specific or narrowly defined topics. It addresses the three key aspects of the information: semantic closeness to search topic, information completeness, and quality. The method is based on algorithms from Natural Language Processing combined with techniques adapted from grounded theory and Dempster-Shafer theory to significantly enhance the discovery of topic related Web-sourced intelligence.

This thesis also describes the development of the new search solution by showing the integration of the mathematical methods used as well as the development of the working model. Real-world experiments demonstrate the effectiveness of the model with supporting performance analysis, showing that the quality of the extracted content is significantly enhanced comparing to the traditional Web-search approaches.

Acknowledgements

I would like to gratefully and sincerely thank my mentor and supervisor Professor David Stupples for his guidance and support. His knowledge, wisdom and commitment to the highest standards motivated me. He also patiently corrected my academic writing. This thesis would have not been possible without him.

I want to thank Professor Martin Newby for helping me on exploring the potential of Grounded Theory, and Professor Sanowar Khan for his general support during my academic life at City. In addition, I would like to thank the School of Mathematics, Computer Science and Engineering for providing me with financial support.

I would also like to thank my husband Dmitry and my parents, who believed in me, inspired me, and supported me throughout the hardest times of this research.

Special thanks to my dear friend Alena for her encouragement, coffee breaks and laugh.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Identifying the Problem

The quality of decisions made in business and government correlates directly to the quality of the information used to support these decisions (McGilvray, 2010). Much of the information used for intelligence analysis may, in the future, be harvested from the Web as this is fast becoming the richest source. This research explored the existing methods of information search and retrieval from the Web as well as from an organisation's locally stored knowledge sources. Three criteria were identified for information that was retrieved by the effective search model – semantic relevance, quality and completeness. The effectiveness of the existing techniques was analysed upon these three criteria. The analysis showed that none of the available methods could effectively retrieve Web information that is semantically relevant to the search topic (that is not just a set of keywords), or provide a metric to identify how good the new piece of information was. In addition, all of the explored methods relied on human judgement to decide how well the new information complemented the already existing knowledge on the search topic. Uncertainty about the amount of information available may result in either information deficit or overload, both of which, in return, lead to poor decision making.

This research explored various Web-based search techniques as well as methodologies that were designed to search across the Intranet held information. Both approaches can rely on statistics or semantics. Statistical Web-based search systems, such as Google, usually incorporate Latent Sematic Indexing (Langville & Meye, 2009) for quick identification of textual items with content similar to the search query. Semantic Web search may involve predefined ontologies (Gruber, 1993) to search across predefined topics or data sources.

As for the Intranet stored knowledge base, relatively smaller amount of non-dynamic information allow semantic search methods to work effectively within enterprise knowledge. This knowledge can also be efficiently searched using enterprise search systems based upon either semantics or statistics. Statistical approaches, such as meaning-based computing (Autonomy, 2008), rely on frequency analysis of terms that correspond to specific concepts. Semantic approaches, similarly to the Web-based search, would also involve ontologies to support query answering against distributed and/or heterogeneous data sources (Stoilos, Grau, & Horroks, 2010).

However, there has not been identified a Web-wide search method that would combine the time efficiency of the Web-based search approaches with the semantic accuracy of the Intranet-based search methods. As originally envisaged by Berners-Lee (2001), the existing Web was expected to form into a Semantic Web, which encourages the inclusion of semantic content in Web-pages, making it both human readable and machine readable. However, Semantic Web is still in its cradle, even though developers have started to publish information using RDF (RDF Working Group, 2004) making it fit for the Semantic Web. Ontologies and typical queries are often fixed at application design time (Stoilos, Grau, & Horroks, 2010) and, thus, put restrictions over the flexibility of the search. Moreover, statistical Intranet-based search technologies require comprehensive automatic clustering, indexing and (often automatic) tagging of the locally stored knowledge base information, making it hard to apply to the big data of the Web.

In order to overcome the above issues, there is a need to systematically identify the semantic relationship between the search topic and the Web-content, and be able to provide information analysts with a quality measure to judge the importance of found information. Moreover, it is important to keep track of the information growth rate due to the risk of information level to become critical and exceed the optimal level, so that it creates the information overload and may lead to the lower decision quality. Therefore, the aim for the analysts will be to collect as much relevant information as possible, thus not exceeding the optimal amount of it and, hence, keeping the decision quality as high as possible.

Information systems are designed to assist human analysts to make judgement of the complex real world using a significantly reduced amount of available information stored electronically. Traditional information retrieval often violates the fundamental law associated with handling complexity – the “Law of Requisite Variety” (Ashby, 1956), which states that only variety can master variety. Applying the law to the exercise of information retrieval from the Web, if the retrieval algorithm does not hold enough variety to deal with the scope and complexity of the Web, its results become attenuated, thus, losing information in the process. In effect, there is a need to identify a search solution for the Web that can handle the vast variety and quantity of information involved, and then to filter relevant Web-pages of high quality, at the same time discarding information that is either irrelevant information noise or too topically remote. However, identifying semantic boundaries of the search topic is not always an easy exercise for information analysts.

Information seekers may come across the situation when they may not have a developed idea of what information they are searching for. Without any knowledge of what information is available for retrieval one is unable to form a suitable query for an information retrieval system. However, a number of studies have shown that although users may have difficulty expressing

exactly the information they want, they are able to recognise relevant information when they see it. According to (Barry and Schamber, 1998), the majority of information retrieval techniques are based on the presence or absence of keywords in relevant documents. However, the reasons why a user may select a document as relevant can depend on many more aspects than simply which terms appear in the document. As indicated by Denos et al. (1997), although users can give explicit reasons for assessing a document as relevant, information retrieval systems cannot use this information to improve a search because they lack the flexibility to detect why a user has marked a document as relevant. This means that traditional keyword search approach leads to missed relevant information since it is impossible to express the "unknown unknowns" related to the search topic in a query.

The concept of the "unknown unknowns" was first introduced by Donald Rumsfeld (2002) in his well-reported speech: "There are known knowns. These are things we know that we know. There are known unknowns – things that we know we don't know. But there are also unknown unknowns. There are things we do not know we don't know." Unknown unknowns make a great impact on quality of decisions as they account for decent limitations and unknown risks, and, hence, should be identified and considered. They reflect implicit assumptions and cannot be addressed due to their implicit nature. In other words, when using an information retrieval system, it is impossible to create a search query that would reflect unknown unknowns because "we do not know we don't know". Thus, decision makers can only rely on their known knowns and known unknowns. And hitherto there has been no process that can measure whether these are sufficient for a good decision to be made.

This thesis addresses this issue and relies on the Grounded theory (Martin & Turner, 1986), (Corbin & Strauss, 2008) that has been successfully used in building a hypothesis (theory) using interviews. As the amount of information, that would be considered enough to build the evidence leading to sensible decision making, is unknown at the beginning of the information retrieval, this research relies on the important characteristic of Grounded theory – it does not require any prior information to generate theory from data.

In order to address the issue of the quality criteria for the information retrieval system, this research utilises Dempster-Shafer theory (Shafer, 1976) that is traditionally used to express uncertain judgements of experts, in this case to measure the quality level associated with gathered information. No manual processing or pre-defining of data source models of any type are required for this theory to calculate the information quality score based on the semantic parameter.

1.2. Research Objective

During the process of decision making one can face uncertainty, unpredictability, or probabilistic nature of the result that is influenced by a number of factors, both internal and external. Combining known methods has the potential to reduce uncertainty through the medium of evidential analysis. The strategy has to be developed that can be used to automatically find and analyse the collected information in order to improve the decision making and decrease the level of uncertainty associated with it.

The objective of this research is to develop an algorithm that supports topic related search for Web-based information and estimates the quality of the extracted content, proving it is significantly enhanced comparing to the traditional Web-search approaches.

This thesis proposes a new approach that supports Web-wide search of topic related information utilising the existing methods from natural language processing for filtering relevant Web-search results, grounded theory to test the completeness of collected evidence and evidential analysis via Dempster-Shafer theory to test the quality of gathered information. High quality information will form a topic-focused knowledge base and can then be effectively manipulated by an enterprise-search engine.

1.3. Research Method

The research method was derived directly from the objective to develop an algorithm that supports topic related search for Web-based information and estimates the quality of the extracted content, proving it is significantly enhanced comparing to the traditional Web-search approaches. The following steps were identified as necessary to meet the objective.

- To investigate the structure of the World Wide Web (the Web).
- To investigate the information gathering process and the ways of information representation in order to ascertain why richness cannot be accessed via a direct keyword based search. Considering the decisions have to be made about the real wold situations, it is important to analyse how information about the real world is stored and accessed, and what effect the limited access to that information may have on decision making. Also, this step would involve exploring the ways of representing and estimating uncertainty on a search topic, as well as how to track the reduction of uncertainty during information discovery.

- To investigate how current search engines operate and to establish why they cannot fulfil a requirement to identify and retrieve information. At this stage, it will be necessary to explore the methods used for the Web search as well as enterprise search, and identify their benefits and limitations, especially when targeted at the discovery of unknown unknowns.
- To investigate research concepts and methods that can be used (or further developed to be used) with a conventional search engine to enrich information retrieval through the discovery of unknown unknowns. Such methods would provide:
  - an effective approach to formulate a topic-focused collection of the Web content that is semantically related to the search topic,
  - a mechanism for tracking the growth of information and the reduction of uncertainty during information discovery,
  - a quality measurement of the extracted content.
- Develop a search model that will integrate a conventional search engine with concepts and methods that will undertake the search. This would need to be a working model that can be tested using the Web.
- Test the model through use cases and undertake analysis of the results. Develop experiments that would address the efficiency of the model to find the Web content that is relevant to the defined search topic, as well as to track the discovery rate of unknown unknowns, and to provide a reliable quality metric that would assess the discovered information from the richness viewpoint.

1.4. Contribution

The main contribution of this thesis is that it addresses the mentioned search issues and presents a novel semantic approach method to Web-based discovery of previously unknown intelligence. It provides a comprehensive theoretical background to the proposed solution together with a demonstration of the effectiveness of the method from results of the experiments, showing how the quality of collected information can be significantly enhanced by previously unknown information derived from the available known facts.

The research objective has been met through the developed algorithm that addresses all three identified criteria of the effective topic-related Web-wide information search. As such, the semantic closeness test relies on the combination of statistical and lexical methods from the Natural

Language Processing. Application of Grounded theory approach allows tracking of the semantic expansion rate of the growing knowledge base and keep it under control. Finally, the proposed algorithm introduces a new quality measure to judge the importance of found information based on its semantic value.

The research objective is addressed through the chapters of the thesis.

1.5. Thesis Structure

1.5.1. Chapter 2

Chapter 2 introduces the nature of information gathering for decision making support and gives background to the subject. This chapter will specifically address the following research method steps:

Investigate the structure of the Web.

Investigate the information gathering process and the ways of information representation in order to ascertain why richness cannot be accessed via a direct keyword based search.

Analyse how information about the real world is stored and accessed, and what effect the limited access to that information may have on decision making.

Explore the ways of representing and estimating uncertainty on a search topic, as well as how to track the reduction of uncertainty during information discovery.

Investigate how current search engines operate and to establish why they cannot fulfil a requirement to identify and retrieve information.

Explore the methods used for the Web search as well as enterprise search, and identify their benefits and limitations, especially when targeted at the discovery of unknowns.

As such, it will address the issues associated with the rapidly growing size of the Web, data verification and data integration. A background is provided to the methods of the traditional information retrieval in both the Web and Intranet. Two fundamental theories are introduced in this

chapter – Ashby's Law of Requisite Variety and Rumsfeld's philosophy, both of which are explained in the context of information retrieval. It describes an individual's knowledge on the topic from the perspective of varying level of uncertainty and provides detail to the conversion of "unknowns" into "knowns" during information discovery.

This chapter also demonstrates the research view of the information flow related to the process of decision making. It is followed by the discussion of current methods of the Web-wide search of topic related information. Among them are statistical methods such as Latent Semantic Indexing, and semantic search methods that rely on ontologies. In addition, it looks at the effective Intranet search methods, which also include semantic approach and statistical methods, such as meaning-based computing, as well as the developing theory of strongly semantic information, which focuses on calculating semantic content. This chapter addresses the issues associated with the above methods of information collection and sharing as well as techniques that have been previously proposed by other researchers in order to overcome these issues.

By the end of Chapter 2 the proposed solution is approved. It will be shown that in contrast to the current research, which tends to focus on specifically selected topics, the new solution has no limitations for Web-sources and is open to perform search and analysis through any Web-page available to the search engine. The proposed new algorithm may be used to harvest Web-wide data in accordance with controlled parameters and subsequently transfer this data to a knowledge base where enterprise search technologies may be applied in the traditional way.

1.5.2. Chapter 3

In Chapter 3 the new algorithm is introduced. It will address the following research step:

Investigate research concepts and methods that can be used (or further developed to be used) with a conventional search engine to enrich information retrieval through the discovery of unknown unknowns. Such methods would provide:

- *an effective approach to formulate a topic-focused collection of the Web content that is semantically related to the search topic,*
- *a mechanism for tracking the growth of information and the reduction of uncertainty during information discovery,*
- *a quality measurement of the extracted content.*

This chapter aims to describe the logic behind the proposed approach and will provide on qualitative rather than quantitative details of each stage of the algorithm. The proposed method consists of the following steps:

1. Identify target knowledge and search objectives;
2. Perform Web-search;
3. Pre-process Web-pages;
4. Perform semantic distance measurement;
5. Build/enhance evidence;
6. Test the evidence quality;
7. Identify further search direction, if necessary.

Steps 4, 5, and 6 represent three stages of filtering and form the major part of the algorithm and incorporate the existing methods from Natural Language Processing, Grounded theory and Dempster-Shafer theory. Thus, for each query, the Natural Language Processing stage filters out only those pages which are semantically relevant to the search topic. Then, the Grounded theory part assesses the Web-pages for holding new information without exhausting the search topic. Lastly, the evidence on the Web-pages is analysed with the Dempster-Shafer theory and its quality level is calculated. It will be shown that such filtering process permits to significantly reduce the length of the initial search results list returned by a traditional search engine, bringing the high-quality documents to the top of the list.

Chapter 3 also describes the fundamental elements required for the proposed algorithm to work. These are as follows:

- initial knowledge base;
- list of queries for search engine;
- Roget's thesaurus;
- list of stop-words;
- search engine;
- Python libraries and script files.

The logic behind the proposed algorithm is that visualises using a rich picture diagram.

1.5.3. Chapter 4

Following the qualitative description of the algorithm in Chapter 3, a detailed quantitative overview of each stage of the algorithm is given in Chapter 4. The aim of this chapter is to address the research step:

Develop a search model that will integrate a conventional search engine with concepts and methods that will undertake the search.

It will address the mathematics behind the proposed algorithm. In particular, this chapter will explain how the initial knowledge base and the collection of extracted texts from the Web-pages are processed as part of semantic analysis. This involves using a hybrid method detailed in (Hirst & Mohammad, 2006), which combines the co-occurrence statistics with the information in a lexical source. At this stage, a text-concept matrix is built for the initial knowledge base and the Web-page extracted text, followed by the Word-Concept Co-occurrence Matrices and Strength of Association Matrices. The chapter will develop a formula for measuring semantic closeness of Web-pages against the initial search topic, using semantic distances between distributional profiles of concepts in two texts.

Chapter 4 will also describe how the collected information is tested for completeness and quality. Grounded theory is used to analyse the amount of new information coming from the Web-pages and to test the completeness of gathered information. This chapter presents mathematics for calculating cumulative amount of new concepts in Web-pages content if they were to be added to the initial knowledge base.

As for the quality measurement stage, the mathematical connection between information retrieval and Dempster-Shafer theory is shown. The fundamental concepts of the theory are put into the perspective of information retrieval. Among them are the frame of discernment, basic probability assignment or mass function, and Belief function, which is used as the quality metric of information in each text from the semantic viewpoint.

By the end of this chapter a clear mathematical model is created, supporting the theory behind the new algorithm.

1.5.4. Chapter 5

In Chapters 5 the process of building the experimental part of the research is explained. The aim of this chapter is to further address the following step of the research method:

Develop a search model that will integrate a conventional search engine with concepts and methods that will undertake the search. This would need to be a working model that can be tested using the Web.

The working prototype is implemented using Python. The aim of this chapter is to give an overview to the model concept and model design, as well as to show how the software was built and tested.

It provides technical details for the fundamental elements of the algorithm described in Chapter 3. UML class diagram for the main module of the software and UML sequence diagrams are presented.

The algorithm presented in this thesis is semi-automated. Chapter 5 gives explanation to the most important pieces of the Python code in the form of pseudo-code. As such, pseudo-code is developed for processing of Google results, extracted text pre-processing and semantic closeness measurement, as well as module with Grounded theory based completeness test and Dempster-Shafer based quality test software modules.

Full program code is available in the Appendix at the end of this thesis.

Chapter 5 also develops model testing on the example of the search topic “cocaine smuggling”. The test query “cocaine production and distribution” was chosen to be focused around the search topic. Precision and recall metrics were used to assess the accuracy of the analysis.

1.5.5. Chapter 6

Chapter 6 provides details of running the experiments, from data collection to model runs. Following the development of model design and software discussed in the previous chapter, this chapter specifically addresses the following step of the research method:

> *Test the model through use cases and undertake analysis of the results. Develop experiments that would address the efficiency of the model to find the Web content that is relevant to the defined search topic, as well as to track the discovery rate of unknown unknowns, and to provide a reliable quality metric that would assess the discovered information from the richness viewpoint.*

The aim of the experiments was to evaluate whether the suggested method significantly improves the process of decision making by extracting only relevant high quality information from the Web.

In total, five experiments were run. Experiments were split in two parts. The first part of the experiments (Experiments 1, 2 and 3) had an objective of assessing the system’s ability to find

and filter Web-pages given a well-defined initial knowledge base and relevant queries. Two topics were used – “tobacco industry” and “cocaine smuggling”, queries were chosen to be semantically focused around the search topic.

The second part of the experiments (Experiments 4 and 5) were aimed at assessing the ability of the system to identify relevant information of high quality for a search topic, assuming that the topic understanding is incomplete and poorly defined. Search topics used were “coffee production” and “tobacco industry”.

The experiment results are illustrated with various graphs showing the following parameters:

- reduction in Web-pages amount after three stages of filtering;
- distribution of semantic closeness values across five queries;
- cumulative KU conversion rate across five queries;
- quality change for top search results.

The chapter is concluded by the table that details and summarises the results of all five experiments.

1.5.6. Chapter 7

Chapter 7 discusses how the research objective was addressed throughout the developed research method. The aim of this chapter is to show that the quality of collected information can be significantly enhanced with the use of the research-developed algorithm. It will outline key research assumptions and findings in accordance with the set research method addressing each of the research method steps and discussing the implications of the research and application of the results presented in the previous chapters. The key research results are summarised at the end of the chapter.

1.5.7. Chapter 8

Chapter 8 discusses possible future work that may improve the performance of the proposed search method. It will explore quality metric of the method and suggest other parameters that may be considered in order to assess quality of a Web-page as a source of information in addition to the semantic value of the textual content.

Web-page content extraction techniques will be discussed that can replace the suggested AlchemyAPI tool and significantly reduce the time of text extraction form Web-pages. This chapter will also discuss how the stop-word list adjustment may affect the performance of the search model.

Code optimisation and overall performance of the search model are explored in order for the created working model to produce results faster with minimised technical requirements.

1.5.8. Chapter 9

Chapter 9 concludes the research and summarises its results.

This thesis introduces a new semantic Web-based search method that automates the correlation of topic-related content for discovery of hitherto unknown information from disparate and widely diverse Web-sources. This method is in contrast to traditional search methods that are constrained to specific or narrowly defined topics. It addresses the three key aspects of the information one or more of which the existing solutions lack: semantic closeness to search topic, information completeness, and quality.

In order to meet the research objective, a research method was developed and addressed throughout the chapters of this thesis.

The research-developed method is based on algorithms from Natural Language Processing combined with techniques adapted from Grounded theory and Dempster-Shafer theory to significantly enhance the discovery of topic related Web-sourced intelligence. It also shows that the new solution overcomes limitations associated with the traditional methods and is open to perform search and analysis through any Web-page available to the search engine.

The list of references is available at the end of the thesis.

Chapter 2

Detailed Description of the Problem

2.1 Overview to Chapter

The aim of this chapter is to explore the nature of information gathering for decision making support at the corporate level within commerce or business, or within the machinery of government. This chapter will specifically address the following research method steps:

Investigate the structure of the Web.

Investigate the information gathering process and the ways of information representation in order to ascertain why richness cannot be accessed via a direct keyword based search. Analyse how information about the real world is stored and accessed, and what effect the limited access to that information may have on decision making.

Explore the ways of representing and estimating uncertainty on a search topic, as well as how to track the reduction of uncertainty during information discovery.

Investigate how current search engines operate and to establish why they cannot fulfil a requirement to identify and retrieve information. Explore the methods used for the Web search as well as enterprise search, and identify their benefits and limitations, especially when targeted at the discovery of unknowns.

The chapter will establish the foundations for the development of the model required for this research. Accepting that a significant amount of information needed for good decision making will be drawn from the Web, this source is expected to increase substantially in the near future. Given that current search engines are not structured to gather increased richness in information, the true value of the Web information may not be realised. The use of ‘richness’ in this context relates to a topic’s true breadth and depth and not necessarily to

priorities of search-engine imposed page priorities. This chapter will therefore delve into the shortfalls of current Web search technology in order to define better the problem that has to be addressed, and hence focusing on the objective of this research.

The chapter will first focus on information gathering for decision making (section 2.2). This section reiterates the need for current and reliable information to support good decision making, and then examines Web data to assess how well it can provide current and reliable information. The section also introduces two key theories which will be used as underpinning tenets for this research; Ashby's Law of Requisite Variety, and Donald Rumsfeld's philosophy on Known Knowns and Unknowns. Nature of information gathering and the scope of the web are discussed in section 2.3.

Section 2.4 will investigate current methods for Web-wide search together with the search within the boundaries of information system for topic related information. The purpose here will be to identify the weaknesses associated with capturing the richness of information required. Important mathematics for analysis used in this research will be identified at this stage.

Section 2.5 will address the issues related to Web-based intelligence gathering where quality and relevance become important. Further mathematical analysis and measurement tools are introduced which will be also required for this research.

The chapter provides the necessary background in the body of this research which is the subject for Chapter 3.

2.2 Information Gathering for Decision Making

2.2.1 The Web as an information source

For individuals with responsibility to be able to formulate good decisions they require current and reliable information (Eppler, 2006). Otherwise the decision made could be flawed. The relationship between data and knowledge and the difference between the terms are important. A good clarification of these terms is given in (Ackoff, 2010). Data represent a set of bytes, numbers, symbols and objects, collected on a daily basis. Organised data are information; they are pre-processed and arranged into structures. Knowledge is integrated information, and includes facts and relationships that have been recognised, discovered or learned.

More broadly, information can be defined as organised data, i.e. data with context added (Michalewicz, 2007). Data can be held in paper form or stored digitally. One of the aspects of successful decision making is being able to access data from a variety of different sources and transform these data into information, and then into knowledge. This thesis will not further clarify the difference between the terms data and information and will use the two terms interchangeably.

The richest source of digitally stored information is the World Wide Web or "the Web" (Fensel, et al., 2006). The Web source can be described as omnipotent or pervasive and grows exponentially in size (The Economist, 2010). It is not only the rapid growth of data that is causing concern. It has become necessary to develop new information retrieval methods since only a small amount of data is structured and much of it is scattered throughout the world. Moreover, the only visible side of the Web is Web-pages. But most of the information is stored behind – in the invisible Web (Barker, 2004), also the access to which is either restricted by privilege or it has to be paid for. Section 2.3 will discuss the structure of the Web and what can be accessed by means of a traditional search engine.

Traditionally decision making have used locally sourced data, i.e. in paper and digital files or 'owned' databases as a basis for decisions. It is estimated that less than 20% of created information is structured and stored in databases (Autonomy, 2009). The rest 80% of digital data in the world is not formally structured and is free format digital content, that is less easily retrievable and usable. Furthermore, databases do not provide users with meaningful information since their main purpose is to store data (Hersh, 2008). To achieve an acceptable degree of integrity the data to be stored is pre-processed to defined structures, but this acts to remove some of the meaning attributed to the data.

However, as business and government become international, locally held data will rapidly become a small subset of the data required for good and reliable decisions. These decisions could embrace commerce, international relations and intelligence. A number of intelligence analysis shortfalls have failed to prevent atrocities with the course be directly attributable to decision makers not having current data available. One of such examples could relate to an accident happened in September 2000 (BBC News, 2000) when the FBI arrested a man who allegedly carried out one of the most devastating financial frauds seen on the Internet yet. The incident renews concerns about the accuracy of information distributed via the Internet. Professional publications are not immune to bad information.

International business is not efficient owing to parochial views being taken rather than from a global perspective. Good managed decisions rely on the information that is

relevant, trustworthy and complete (English, 2009), meaning that the information has the required scope and depth (richness) for the decisions to have integrity. Generally, information stored as data in organisation's owned databases lacks scope as storage is limited and historic data gathering may not have functionality for an effective gathering task (Zailani & Rajagopal, 2007).

The Internet currently requires the support of human data gathering. Relevant and trustworthy information required for a particular decision using the Web is difficult to find since the rest majority of it fails to fit a given requirement. This can be demonstrated by examining a few basic facts. According to Netcraft statistics, as of July 2010 the number of responsive/active Web sites was 205,714,253 (Netcraft, 2010). Four years later in July 2014 the number of sites grew up to 975,262,468 (Netcraft, 2014). Figure 2.1 below demonstrates growth trend in both active sites and, especially, reserved hostnames.

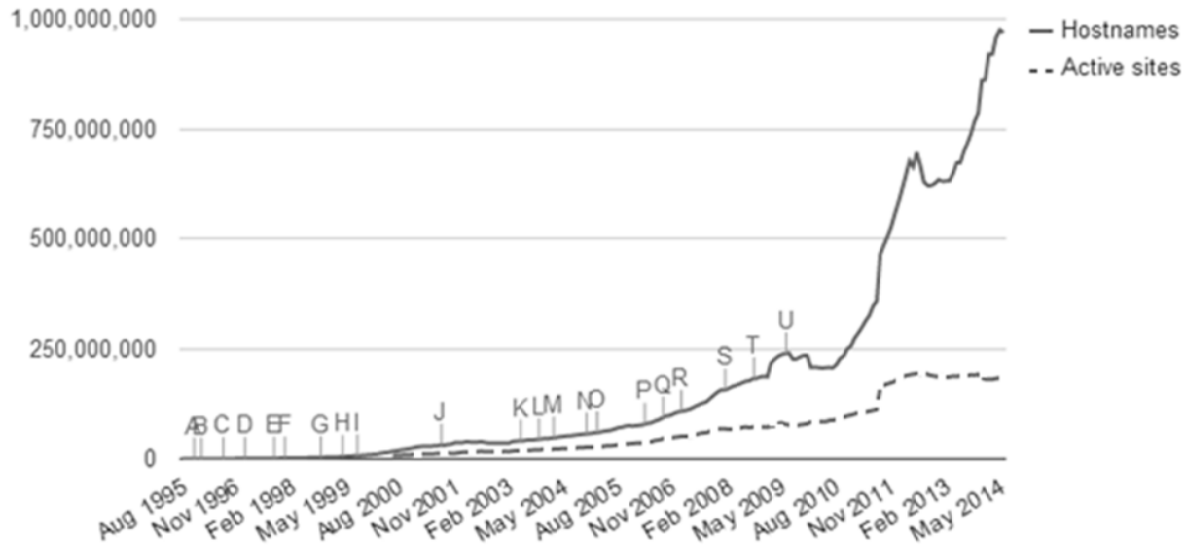


Figure 2.1 – Netcraft reserved hostnames statistics

With such a rapid expansion of available Web data, evaluating trust in it has become an important issue. Even though the Internet gives access to a huge amount of information, it is becoming important to discriminate: data can be duplicated, highly contradictory, contain errors and exaggerations.

In addition to the growth of the Web (Netcraft, 2014), data verification and data integration is another important issue. Organisations tend to store their information in many disparate data sources. The same can be said about the Web. Decision making can be

seriously affected by insufficient or unavailable access to information located in these non-integrated, redundant, stove-piped data sources. Effort should be made to use an integrated system that would provide a solid, stable, but flexible platform for effective information processing.

To reiterate, in this modern world both business and research require decisions based on current, relevant and trustworthy information (English, 2009). However, information required for a particular decision is difficult to find using the Web since the majority of it fails to fit these requirements. While there are no quality control procedures for information uploaded, data remains duplicated and highly contradictory. Around 80% of data is stored in unstructured format (Holzinger, et al., 2013). Selection of data for a decision information gathering task can be complicated and sometimes lead to the wrong decision making, since it is getting harder to find relevant information of high value and there is no obvious way of measuring the quality of information gathered. The graph in the Figure 2.2 illustrates how the quality of decisions depends on the amount of information these decisions rely on.

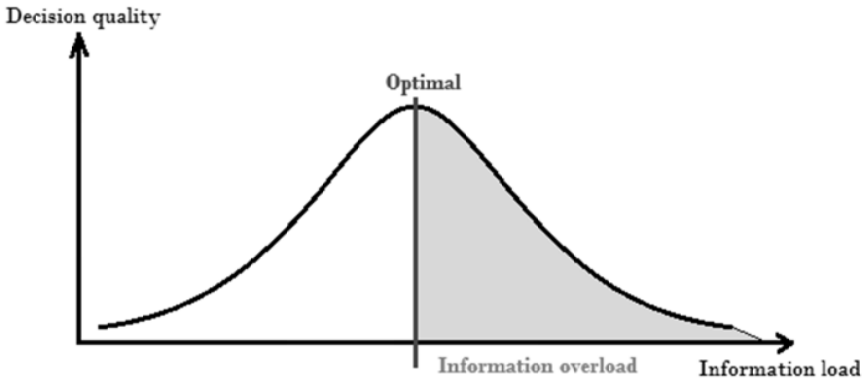


Figure 2.2 – Information overload effect on decision quality

As long as the knowledge on the topic is growing, the increasing amount of available information positively affects the decision making since more details can be considered. However, at some point the amount of available information surpasses the optimal level causing information overload which refers to the difficulty an individual can have understanding an issue and making decisions that can be caused by the presence of too much information (Yang, Chen, & Hong, 2003). It is highly important to correctly formulate

the emerging issues and identify possible options for solutions. Effective decisions not only cost less, but also they lead to fast and better results.

At this point Ashby's Law of Requisite Variety and Rumsfeld's Philosophy are introduced in order to develop base theory for this research.

2.2.2 Ashby's Law of Requisite Variety

The idea of information retrieval systems often misses the fundamental laws associated with handling complexity. One of them is Ashby's Law of Requisite Variety. In the middle of the 20th century William Ross Ashby formulated the fundamental «Law of Requisite Variety» in his book «An Introduction to Cybernetics» (Ashby, 1956). In cybernetics the term variety denotes the total number of distinct degrees of freedom or states of a system. It was introduced by Ashby to define the count of the total number of states of a system.

If there is a certain system, it has an internal structure, but there are also external conditions. A variety of a system is a set of its conditions, possible as a result of influence of an environment, and action of the internal reasons. And the life of this system is caused by interaction of a variety of an environment and an internal variety of system.

According to Ashby, the Law of Requisite Variety states that only variety can master variety, reducing disturbances and promoting harmonious order. When the law is applied to the Web search exercise, it becomes clear that the retrieval algorithm may not be amplified enough to deal with the scope and complexity of the Internet, and its results appear to be attenuated in an ad hoc manner, thus, losing information in the process.

If variety of the environment is smaller than variety of the system, the system starts to adapt its environment by exporting its variety outside. If variety of the environment is larger than variety of the system, the system itself needs to be arranged under the advance growth of complexity of the environment, but this is not always possible. In this case the system fails or becomes controlled by another system. In order for the system to remain stable, the number of states of its control mechanism must be greater than or equal to the number of states in the system being controlled. In other words, the system should continuously support such level of internal variety which would correspond to variety of the environment. Only in this case the system is capable of keeping a balance with the environment.

To put this further in context for information systems an example is provided. When an information system is first designed for a business, the variety of the business will equal the variety of its business environment. As the business environment changes (in the real world this change could be rapid) the variety of the business environment can rapidly exceed the variety of the designed information system. Such change would require the information system to be redesigned in order for it to be able to balance the complexities of the business and the controlled environment; however, such change cannot normally be achieved rapidly enough.

Moreover, if the Internet as an information source is considered to be a network with a floating number of elements, there is no limit to the information that can be held. It will never match the amount of the information from the real world, where the decisions are made and where such decisions affect the real world. This research considered the question of variety during the development of retrieval approach.

2.2.3 Rumsfeld's philosophy

The former USA secretary of defence Donald Rumsfeld said in his speech (Rumsfeld, 2002): "There are known knowns. These are things we know that we know. There are known unknowns – things that we know we don't know. But there are also unknown unknowns. There are things we do not know we don't know." It is the last one – unknown unknowns – that is of the highest value. Known unknowns (KU) reflect explicit assumptions and therefore can be addressed. Unknown unknowns (UU) reflect implicit assumptions and cannot be addressed due to their implicit nature. When summed up both KU and UU reflect the uncertainty.

This research assumes, that considering an individual's current knowledge, any topic on any subject can be split into KK, KU and UU. The ratio of all three will be different for different people. The diagram in Figure 2.3 schematically shows how an individual's knowledge on a subject can be split into KK, KU and UU.

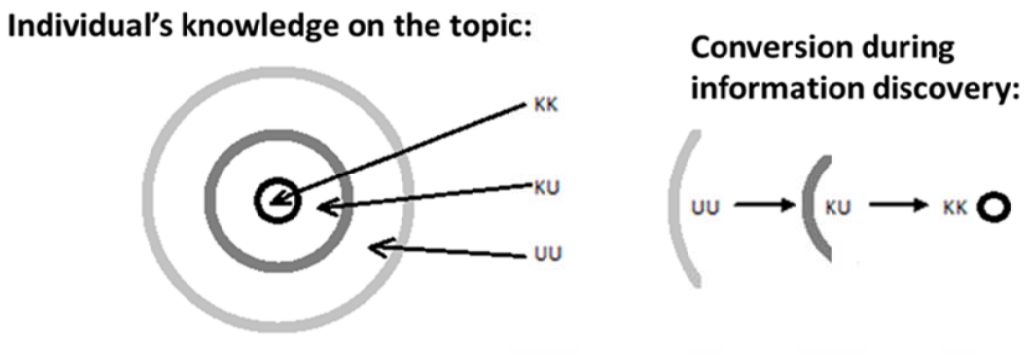


Figure 2.3 – Rumsfeld's terms in information discovery

In the above diagram the inner circle represents an individual's KK, while the thick border of it is the KU. The outer circle area is the UU. The more this person learns about the topic, the bigger the area of KK circle becomes, and at the same time, the larger the enclosed area of KU is. Hereby, information changes its state from UU to KU until it becomes KK. It also may be that as soon as UU is discovered it becomes a KK.

The process above closely relates to change of uncertainty proportion within the individual's knowledge during the process of information discovery (figure 2.4). It assumes that for every subject the total knowledge on this subject is the same for every person, it is the amount of uncertainty (that resides in KK and KU) that is unique and depends on the depths of an individual's knowledge on the subject.

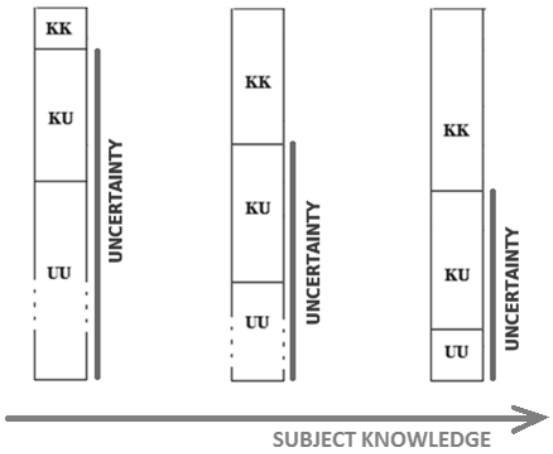


Figure 2.4 – Knowledge expansion

An individual, who is new to a topic or has very little knowledge about it (left pillar), will have only a few subject related KK. A slightly larger amount of KU exists, while

the UU pool will be the bigger part of the total knowledge on the subject. In this case, the uncertainty is much greater that his knowledge.

With discovery of new information (middle pillar), the KK circle grows in size pushing the uncertainty level down. Moreover, it creates changes in the structure of uncertainty – with expansion of the KK the amount of KU grows reducing the level of UU. The right pillar demonstrates the changes that relate to discovery of a significant amount of new information which makes an individual's knowledge on the topic close to exhaustion.

The ideal model would correspond to the uncertainty level approaching zero, i.e. all the UU are discovered and converted to KU and KK. Such conversion of "unknowns" into "knowns" is the main purpose of information discovery and is one of the major ideas that lie behind this research.

2.2.4 Conversion from KU into KK

Donald Rumsfeld (2002) defined the terms "known knowns" (KK), "known unknowns" (KU) and "unknown unknowns" (UU). Effective decision-making requires trusted, focused and relevant information. We should be comfortable with both KK and KU, as these are straightforward to find. The problem being that much of the rich information required for good decisions may be in the category of UU. So the important question to be asked is how we find the relevant UU to enrich and improve decision-making? In effect there is a need to identify a search solution that can handle the vast amounts of information involved and in the very many different format types. Such solution is the subject of this thesis.

Not all the information from the real world becomes stored. A part of the real-world representation can be seen as held within the global knowledge base. Noteworthy, this part is much smaller than the rest of the information left behind. The global knowledge base can only store information about the events in the real world that are observed by various sensors (human, technology). The rest of events remain unknown.

Figure 2.5 briefly demonstrates the relationship between the real world and an individual's knowledge during the process of information discovery.

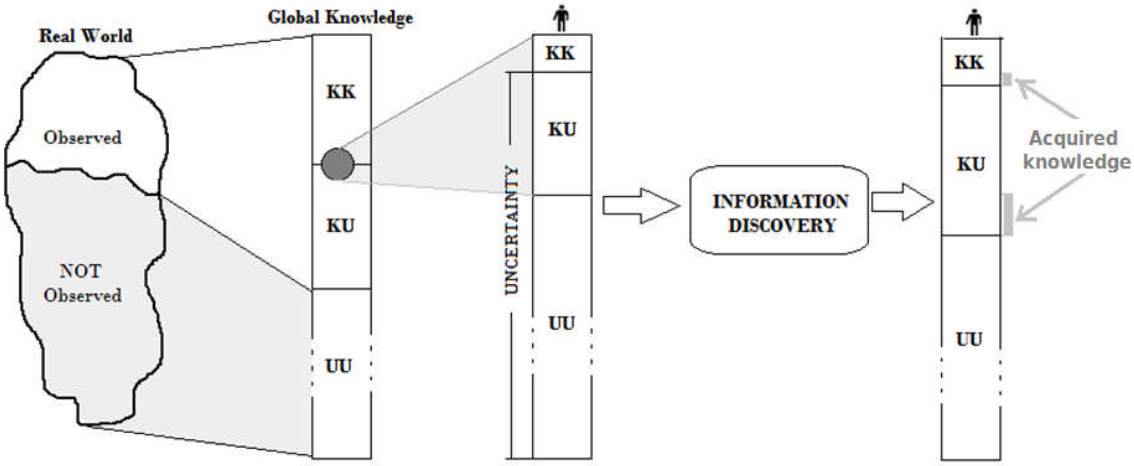


Figure 2.5 – The real world and an individual's knowledge

In order to make a decision about the real world an individual will have only a limited access to the information stored in the global knowledge base. That is, for example, the individual will be able to talk only to people who can be contacted, or browse the information online that is seen by a search engine. However, the individual will never be able to get access to all the information since only part of it is replicated. Therefore, the individual's knowledge on the topic of the decision is limited to the information found and, thus, the individual will have to amplify that limited information (according to the law of requisite variety) and produce new knowledge to make a decision about the real world or affect the real world. What is important, if the information acquired is of poor quality and there is still a large part of uncertainty uncovered, the effects on the real world can be serious and dangerous.

2.3 Nature of Information Gathering

2.3.1 Scope of the Web

The world holds a vast amount of printed and digital information. Information can be broadly defined as organised data, e.g. data with context added (Michalewicz, 2007). Quantifying the total amount of information that exists in the world is hard, but it is clear that it grows exponentially every year (The Economist, 2010). The issue of information

architecture was raised by (Morville & Rosenfeld, 2006) highlighting the importance of being able to get the right information fast.

A vast amount of information is a form of virtual representation of real world situations. It is important to understand that real world is changing all the time, and human use sensors of various types to gather data and information and store in a global knowledge base. The Global knowledge base is persistent data, that is the source of information for any report that is written and any decision is made.

Modern world generates large amount of information in every moment of time. One part of information remains unknown just because there was no record made about it. The other part stays in the Global knowledge base, formed of people's memories, printed material, digital world, etc. While the real world is dynamic and changes all the time, the global knowledge base is persistent. Obviously, the set of all recorded information is just a reflection of the real world and contains only a fraction of total information produced.

It is noteworthy that, if the decision has to be made about the event in the real world, the only information an individual can rely on will be stored in the global knowledge base. And for this decision to be as good as possible, the individual would have to form up a set of problem related information of highest quality. So that when the gathered information is amplified to match it with the real world, according to Ashby's Law of Requisite Variety, there are fewer mistakes during the amplification process, hence, the higher chance of correct decision making.

The rich picture below (Figure 2.6) demonstrates in schematic form the information flow related to the process of decision making. This thesis will focus on the content of the World Wide Web, as a part of digital set of the global knowledge base.

When an individual forms a query about the situation observed in the real world or the Web, the information being used also goes through attenuation, and upon his best knowledge the individual would create a set of keywords for a search engine that would potentially reflect the search topic. Moreover, an individual who is going to analyse this data to write a report to a decision maker will do so based on the data from the global knowledge base. Section "2.4.2 – Searching the Web" will look at the query and search mechanism at a more detailed level, while section "2.4.3 – Effective Intranet Search" will explore the techniques behind searching within the boundaries of a locally held information source.

According to Ashby and his law of requisite variety (Ashby, 1956) it will be impossible to find the desirable information about the query on the Web, unless it has been recorded there by a real world observer. Thus, it is more likely to find an information online

about something that was taken from a published source (book, radio, TV, etc), rather than something that was observed by somebody in the real world, because the interpretations of the event can be different or misinterpretation may take place. That is why search on a complex query often fails – it is quite difficult, even impossible, to predict how information about a situation was transformed by the publisher, we do not know what we do not know.

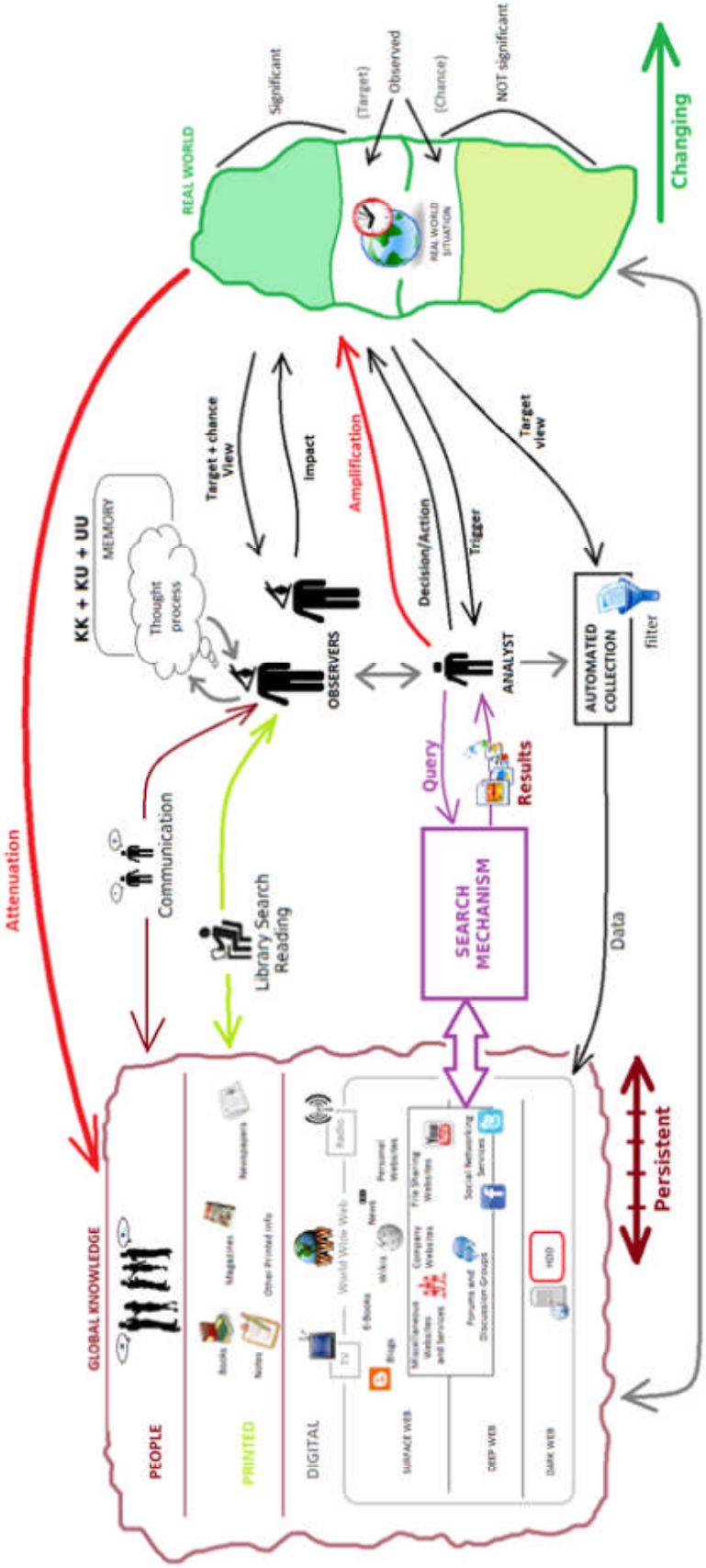


Figure 2.6 – Information flow related to the process of decision making

2.4 Information Discovery from the World-Wide Web

2.4.1 Finding information on the Web

Well managed decisions rely on information that is relevant, trustworthy and complete (English, 2009). Managed wisely, information can be used to unlock new sources of economic value or provide fresh ideas for scientific research. Information is everywhere and it grows rapidly every day. The Web has already become the key information source for the future. Nevertheless, finding useful information is often a frustrating experience. There has been a need for new effective techniques to perform an intelligent search and discovering useful information.

Over many years information managers have used database management systems to search and store vast amount of data, with the idea that these data have some value. However, it soon became clear that data itself does not have great value (Zailani & Rajagopal, 2007). It is the ability of an organisation, to wisely analyse the available data in order to add value. This means a business intelligence system should be able to access data from a variety of different sources and subsequently transform these data into information, and then into knowledge. This allows better decision making, and it is not surprising that better business decisions usually translate into better financial performance.

Despite the fact, that during the last decade IT specialists have put expended considerable to organise their data into formats prescribed by enterprise resource planning (ERP) systems (Monk & Wagner, 2009) and various structured databases, more than 80% of data remains stored in unstructured data files (Autonomy, 2009). This unstructured data could comprise MS Word or PDF format documents, e-mails, recorded phone calls etc. Database Management Systems search tools are efficient, if users know what to look for using key words or phrases. However, such tools are strictly limited to the schemas being used to structure the data (Beynon-Davies, 2004). It must be emphasised that it is not information that is normally stored, but data. These data become information only when a user/analyst interprets the search results.

The traditional approach to the Web search engine, such as Google, is based on indexing of the Web-content, building the index database, and then searching for the keywords that match the content of this database. At first sight, it seems convenient and it has

been working fine in Internet for many years, perhaps giving the illusion that the search engine is searching for the content you require.

Over the last few years, enterprise (Intranet) search techniques have been developed (Delgado, et al., 2005). Predominantly they are based on building an 'index database' over the whole enterprise Intranet. Among the most widely used search systems is a product called Autonomy (Autonomy, 2009). Autonomy employs Bayesian statistical models, similar to those used to filter spam, to determine the categories of documents, and then is builds an index of the text pattern. Smartlogic (Smartlogic, 2001) uses ontologies to turn enterprise content into a semantic Web. However, there are issues associated with the existing information discovery methods from both the Web and Intranet. Such issues will be discussed towards the end of this chapter. First, it is important to understand what lies behind searching the Web.

2.4.2 Searching the Web

Usually, “the Internet” and “the World Wide Web” (the Web) are considered to be the same thing. However, they are totally different things and it is important for this research to understand the difference. The Internet is a global data communications system with its hardware and software infrastructure providing connectivity between computers to form a global network. The Web is one of the Internet services, a network of interconnected documents and other resources, linked by hyperlinks and URLs. A hyperlink is a reference to data that a user can follow or that is followed automatically. A hyperlink usually points to a whole document or to a specific element within a document. Hypertext, on the other hand, is text with hyperlinks. A uniform resource locator (URL aka known as a Web address) is a specific character string that constitutes a reference to a resource.

Obtaining knowledge, not information, from the Web is a significant issue. Not just because most of it is hard to reach employing a search engine, but because the information on the Web-sites is stored in silos – such an approach was supposed to boost the performance of search engines as it helps to ensure the keyword focus is not diluted. Even so, the desirable level of information quality brought back by a search engine is questionable, as search engines look down the silos, not across them. However, the Web is still the richest source of digitally stored information. According to Cisco, annual global IP traffic will surpass the zetabyte (1024 exabytes) threshold in 2016 (Cisco, 2014).

The Web is a distributed repository linked by millions of hyperlinks embedded in hypertext documents. This metadata makes it possible to efficiently find the most relative documents. Before proceeding with this discussion it will be helpful to identify the layers of the Web in order to be able to visualise where data is located.

There Web can be split into three layers: Surface Web, Deep Web and Dark Web as shown in figure 2.7.

Figure 2.7 – Three layers of the Web

The Surface Web (Shikha, et al., 2009) is a part of the Web that is reachable by a search engine; it is the smallest portion of the Web. According to Netcraft's monthly survey, as of July 2014 there were 975,262,468 active Web sites (Netcraft, 2014). Google has the largest index base among the search engine fraternity and knows only less than 1% of the total Web. Deep Web (Raghavan & Garcia-Molina, 2001) refers to a part of the Web, that traditional search engines cannot "see" or retrieve content from. Google has access to a small part of the Deep Web (Madhavan, et al., 2008). Dark Web (Chen, 2011) stands for Websites and secretive networks that sometimes span across the Internet and cannot be accessed through conventional means. Usually access to the Dark Web is via an anonymity system such as 'The Onion Router' or TOR (Li, et al., 2011).

The rich picture at Figure 2.8 shows a traditional view of information gathering using a Web-search engine and an enterprise search system.

As has been previously stated, the Web may be considered as a virtual representation of the real world, consisting of a combination of KK, KU and UU from an individual's perspective. When searching for a piece of information an individual relies on the information known in order to form up a search query. The query is formed based on limited

information about the real world – resulting from an attenuation process as per Ashby's Law of Requisite Variety.

When sent to a Web-search engine, the query string progresses through the chain of comparison processes returning a list of search results to the user. The next section "Statistical Methods – Latent Semantic Indexing" provides details on how Web search engines work using Google as a use case.

Having analysed the Web search results an individual makes a decision on its relevance to the search topic and, if necessary, the result can be used to populate a locally held knowledge base (maybe intranet based) with the newly found information. This knowledge base may contain information from a previously targeted search and hence it is likely that knowledge on this topic is being enhanced.

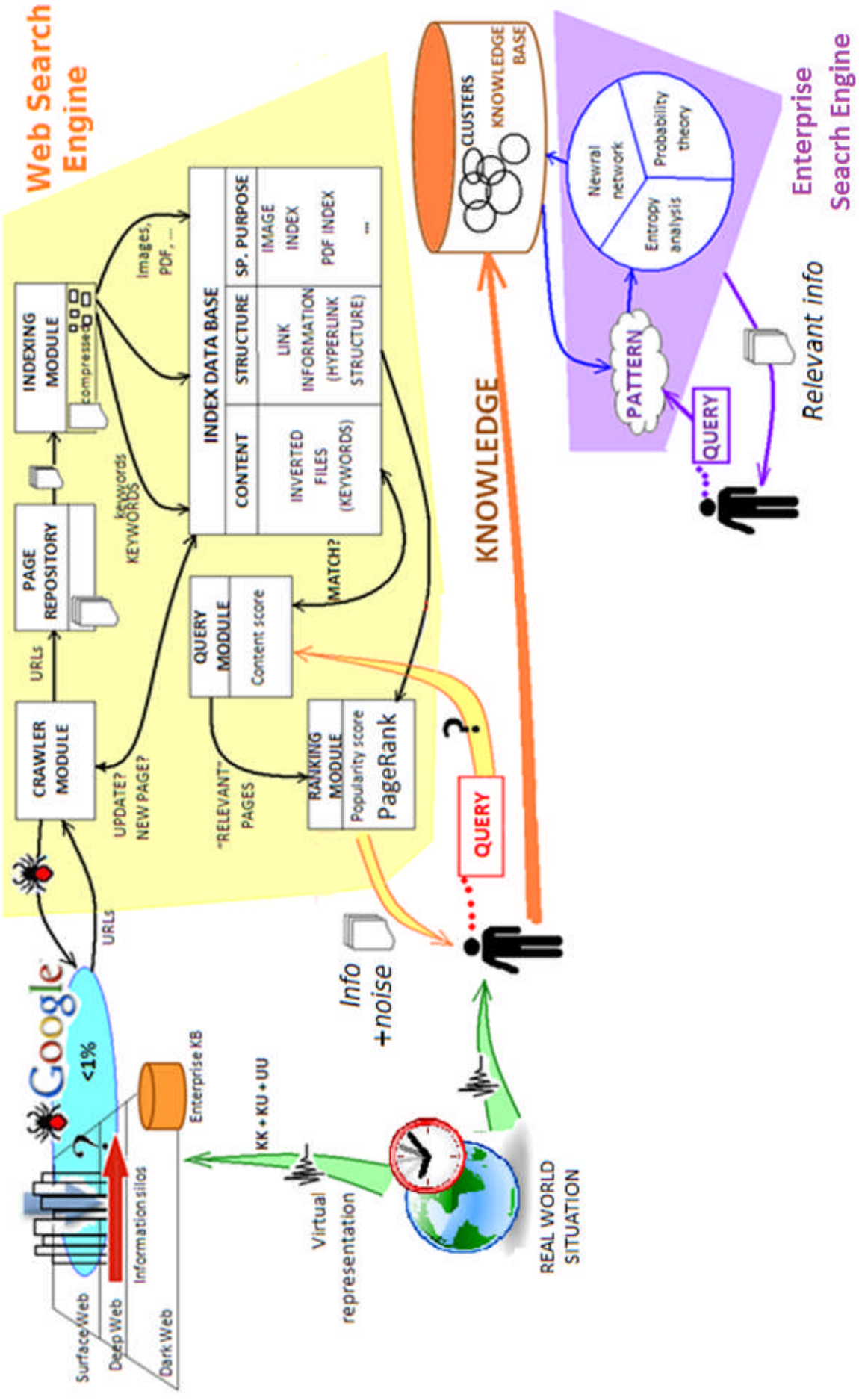


Figure 2.8 – Traditional view of information gathering using a Web-search engine and an enterprise search system

It is also possible to perform some complex processing on this knowledge base possible searching across intranet files to gain value. The section 2.4.3 addressing "Effective Intranet Search" will explore concept search methodologies that can be currently used for search within an enterprise knowledge base. Concept search techniques are more effective than those based on keyword search for finding topic related content, but they have limitations with regards to the size of knowledge base.

2.4.2.1 Statistical Methods – Latent Semantic Indexing

All Web search engines follow the same algorithm: Web-crawling >> Indexing >> Searching. Google is the most popular Web search engine. It uses automated programs called spiders and has the largest index of keywords and where those words can be found.

A detailed explanation of how Google search engine works is presented in (Langville & Meye, 2009). This section presents a brief overview to Google's modules and Web-search processes in general terms to a level that will satisfy the research objective. The rich picture (figure 2.9) below demonstrates the connections between different parts of the Google search engine.

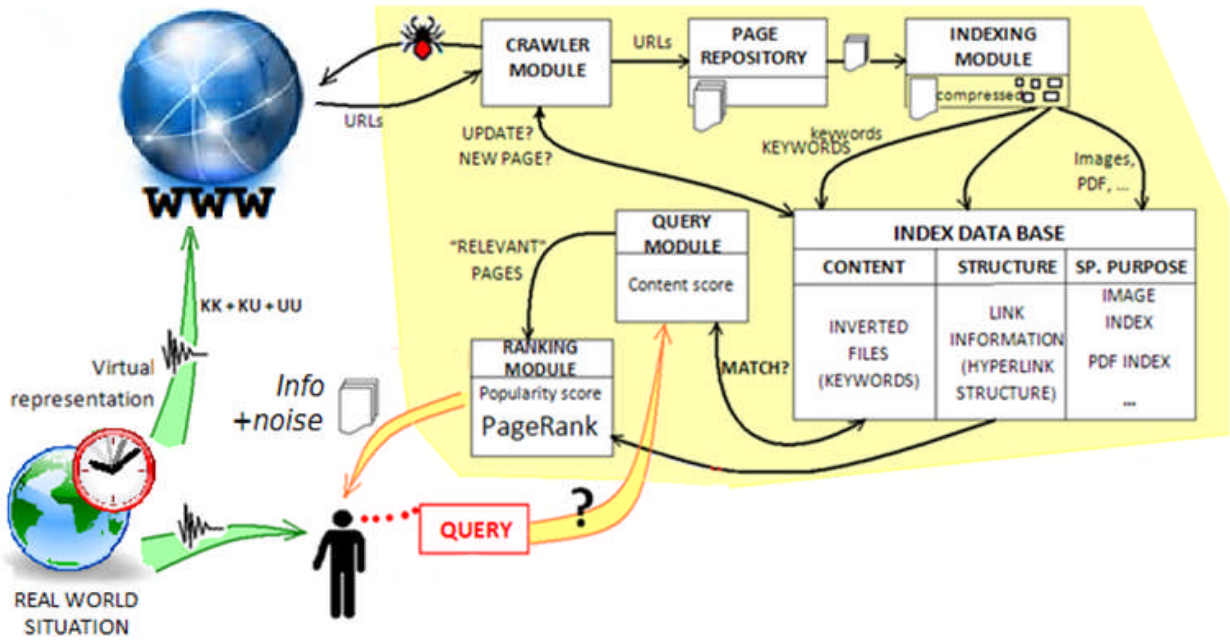


Figure 2.9 – Overview to the Google Search Engine

Firstly, the crawler module uses spiders for crawling the Web by giving a spider a set of URLs to visit. Spiders start on pages addressed through the URL and then follow all the hyperlinks on those pages to find new pages. Spiders return with new or updated pages which are temporarily stored in a page repository.

The indexing module takes every page from the page repository and creates a compressed description of the page by extracting vital descriptors and storing them in specific indexes. The indexing module pays attention to information in title, any associated description, anchor texts, hyperlinks, and terms in bold or large font. Specifically:

- Structure index contains valuable information regarding the hyperlink structure in the search engine. Crawlers sometimes go to the structure index to find uncrawled pages.
- Special purpose index holds information about images, or pdf that can be useful for a particular query task.
- Content index stores textual information on the page in a compressed form using an inverted file structure.

For example:

term 1 (*apple*) – 2[1, 1, 27], 128[1, 0, 7], 1598 [...]
term 2 (*arc*) – 2[...], 12587[...]
...
term 15 (*juice*) – 2[1, 1, 10], 56[...], 128[0, 0, 5], 1025 [...], 30214[...]
...
term *m* (*zoo*) – 8[...], 548894522[...]

This means that “term 1” is used in pages 2, 128 and 1598. In the vector [1, 0, 7] “1” means that term “128” appears once in the title tag of the page 128, “0” means it does not appear in the description tag of the page, and “7” is the number of times term “128” appears in the page body. Such a three-dimensional vector is inserted after each page identifier for every term.

An advantage of an inverted file is that it can be used as a quick look-up table. However, because the number of “term *m*” is huge, the file size is huge. Also, there are a large number of pages that use popular broad terms. Moreover, page identifiers are not the only descriptors stored for each page. Therefore, indexing process consumes storage and

needs a clever strategy to keep index database up-to-date. In 2010 Google introduced the Caffeine index (Grimes, 2010), the image below (figure 2.10) illustrates the logic of how the old indexing system worked compared to Caffeine.

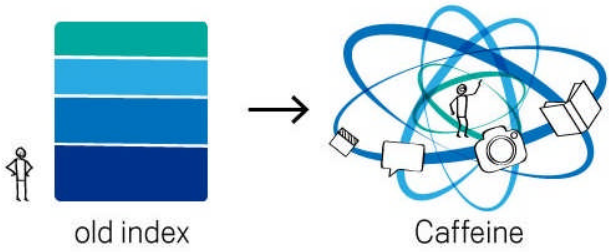


Figure 2.10 – Google's Caffeine index

The old index had several layers, some of which were refreshed at a faster rate than others; the main layer would update every couple of weeks. To refresh a layer of the old index, the entire Web would have to be analysed causing significant delay for the page to become searchable. With Caffeine, the Web is analysed in small portions allowing the search index to update on a continuous basis and on an enormous scale.

When a user submits a query, the query module accesses the indexes in order to answer the query. At this stage, a search engine compares the keywords in the query string with the list of terms in the content index and calculates the content score of the pages that contain the keywords.

For instance, a user entered the query of "*apple juice*", assuming that Boolean AND is used. The query module looks at the inverted list of terms for *apple* (term 1) and *juice* (term 15). The pages that use both query terms are 2 and 128:

term 1 (*apple*) – 2[1, 1, 27], 128[1, 0, 7]
term 15 (*juice*) – 2[1, 1, 10], 128[0, 0, 5]

To calculate the content score of a page, one needs to add the values in the three-dimensional vector for the page and multiply it with the sum values for the same page for every term.

Thus, the content scores (CS) for the two relevant pages are:

CS(page 2) = (1+1+27) x (1+1+10) = 29 x 12 = 348
CS(page 28) = (1+0+7) x (0+0+5) = 8 x 5 = 40

The query module is query-dependent. The query module passes the set of relevant pages to the ranking module. The content score is calculated from the Term Frequency – Inverted Document Frequency (TF-IDF) matrix (Rajaraman & Ullman, 2011), where columns represent all the Web pages the crawler knows and rows are all the keywords on these Web-pages. Cells contain the weight indexes for the respective word in the respective Web-page.

The general formula for a keyword weight index is:

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i}) \quad (2.1)$$

where $\boldsymbol{tf_{i,j}}$ – the number of occurrences of a word *i* in a document (Web-page) ***j***,

$\boldsymbol{df_i}$ – the number of Web-pages containing ***i***,

N – total number of documents

It is clear that TF-IDF matrix is extremely large. Google uses stemming, morphology and synonyms to relate the keywords to the words in documents. The term stemming (Lovins, 1968) is used in linguistic morphology and information retrieval to describe the process for reducing inflected (or sometimes derived) words to their stem (sometimes referred to as their root). Google also uses mathematical transform techniques and matrix decomposition in particular to manipulate the tables. It is data-driven and also global in nature, which means this technique is capable of much more robust information extraction and representation of semantic information than techniques based on statistics.

Latent Semantic Indexing (LSI) (Deerwester, 1988) is an indexing and retrieval method that uses common linear algebra to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. The method constructs a weighted term-document matrix, where rows represent terms and columns represent documents in the collection. The process involves singular value decomposition on the matrix, and then uses the matrix to identify the concepts contained in the text. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts.

As it uses a strictly mathematical approach, LSI is language independent. This enables LSI to extract the semantic content of information written in any language without the use of any dictionaries and thesauri. Moreover, LSI automatically adapts to new and changing terminology, and its performance is not affected by noise, i.e. spelling mistakes, unreadable characters, etc.

The query module is query-dependent. It passes the set of relevant pages to the ranking module.

The ranking module sorts the set of relevant pages according to some criteria. It employs the information stored in the structure index of the search engine to calculate the popularity score. Google uses a PageRank algorithm to order a list of pages and place the most relevant closer to the top. In contrast to the query module, the ranking module is query independent, and provides a global ranking of all pages of Google's index which covers over 45 billion Web-pages as of October 2014 (www.worldwidewebsize.com).

PageRank is a solution to a linear equation (Chiang, 2012). According to the PageRank strategy, “a page is important if it is pointed to by other important pages”. The PageRank of a page *Pi* is the sum of the PageRanks of all pages pointing into *Pi*. It is an iterative procedure.

The combination of the content score and the PageRank (popularity) score gives an overall score for each relevant page. The result of the searching process is a list of Web-pages, organised in a special way. The first line a user will see is the title. That is what is set in the title of the Web page. Under the title there is a description of the page, called a snippet. It can come from different places. It can be taken from the meta-description tag. The description can also be pulled directly from the place within the page. If Google can get nothing from the page, not even a meta-description tag, it is taken from the Open Directory Project also known as DMOZ – largest, most comprehensive human-edited directory of the Web (DMOZ, 2014).

As a result, Google is able to deliver millions of results for many of its queries in a fraction of a second. However, quantity does not always mean quality. Since the calculation of the content score involves matching two sets of keywords, the part of the relative quality information is overlooked (it even may be all the quality information), explained with different words. Also, as the ranking module of Google's search engine is query-independent, it does not bring the results that are most relevant to the user, but the pages that other people decided to link to. Pages with high quality (for example, scientific) information may not be popular, while popular documents may not be relevant to the search query. This leads to often unwanted advertising pages come first, moving highly semantically relevant documents to the bottom of the list.

The big disadvantage of Google’s PageRank algorithm is that it doesn’t give the results that are most relevant to the search idea, but the ones that enough people consider worth linking to. High quality (scientific) documents may be overlooked because they are just

not popular. On the contrary, popular documents may be responsive but not relevant to the search query.

To sum up, the Web has provided means of retaining the original scattered unstructured data by using search engines such as Google to locate and process meaningful information. When a user does a Google Web-search, he is not searching the Web itself, but Google's index of the Web – billions of Web pages stored across thousands of servers. After getting a query string, Google software starts searching the index to find every page that includes the search terms. From hundreds of thousands of Web pages indexed, Google finds those most "relevant" with the use of the PageRank algorithm. The formula, that rates the importance of the Web page, looks at the frequency and locations of the keywords within the Web page, how long the Web page has existed, the number of outside links pointing to the page and how important those links are (Langville & Meye, 2009).

The problem with this approach is that the majority of data that could enrich an information response is missing since the search engine has either not located the specific keyword in a document or has prioritised wrongly. This happens because of the effects of synonyms and polysemy. In addition, keyword search usually excludes misspelled words. Even though Google constantly improves the logic behind its search algorithm to deal with such tweaks, it is still far from being perfect. This inability to understand information means that other documents that present the same idea but use different words are often overlooked. Equally, documents with a meaning entirely different to that which the user searches for are frequently returned, forcing the user to alter their query to accommodate the search engine.

2.4.2.2 Semantic Search - Ontologies

Evidence can be used as an informational asset and to the inferential process that transforms it into knowledge (Laskey, Costa, & Janssen, 2008). Practitioners from many disciplines can profit from a formalisation of the discipline of evidential reasoning. Due to its heavy dependence on evidence in almost every aspect of its operations, the domain of intelligence analysis would be a prime beneficiary of ontology of evidence. Benefits of ontology of evidence include a common, shared vocabulary for important features and relationships that occur across different applications of evidential reasoning, as well as the ability to share information among diverse systems.

Despite considerable diversity and individual variation in the conduct of investigation and analysis, there are fundamental common structures and processes. A formal representation of evidence and evidential relationships provides the obvious benefit of allowing analysts to query a knowledge base for the evidence on which the conclusions are based.

There has been an increasing emphasis in recent years in sharing knowledge among intelligence applications, and ontology of evidence and inferential reasoning can be helpful. Ontologies (Gruber, 1993) provide shared representations of the entities and relationships characterising a domain, into which vocabularies of different systems can be mapped so to provide interoperability among them. Shared formal semantics enables systems with different internal representations to exchange information, and provides a means to enforce business rules such as access controls for security.

However, traditional ontologies do not provide a principled means to ensure semantic consistency with respect to issues of uncertainty related to credibility of sources, relevance of evidence, and other aspects of the evidential reasoning process (Laskey, Schum, Costa, & Janssen, 2008). Because uncertainty is a fundamental aspect of evidential reasoning, this is a serious deficiency.

When faced with the challenge of representing uncertainty in ontology, the natural tendency is to introduce a means to annotate property values with information regarding their level of confidence (Costa & Laskey, 2006). Over the past several decades, semantically rich and computationally efficient formalisms have emerged for representing and reasoning with probabilistic knowledge (Laskey K. , 2008).

Search engines are very effective at filtering pages that match explicit queries. Unfortunately, most people find it difficult to express what they, especially if forced to use a limited vocabulary such as keywords. The result is large lists of search results that contain a handful of useful pages, defeating the purpose of filtering in the first place. There are two major approaches to user modelling – based on knowledge and based on users behaviour (Fischer, 2001). Knowledge-based approaches create static models of users and dynamically match users to the closest model. Behaviour-based approaches use the user's behaviour itself as a model, often using machine-learning techniques to discover useful patterns of behaviour. Ontology can be used to investigate how domain knowledge can help in the acquisition of user preferences.

A step forward in customising Web-search has been done with introducing ontologies in recommender systems. In (Middleton, Shadbolt, & De Roure, 2004) the authors

explore a novel ontological approach to user profiling within recommender systems, working on the problem of recommending on-line academic research papers. Two experimental systems are introduced – Quickstep and Foxtrot. Both systems create user profiles from unobtrusively monitored behaviour and relevance feedback, representing the profiles in terms of a research paper topic ontology.

Quickstep (Middleton, De Roure, & Shadbolt, 2001) is a content-based recommender system that uses ontological inference to improve profiling accuracy and integrates an external ontology for profile bootstrapping. It monitors user browsing behaviour via a proxy server, logging each URL browsed during normal work activity. A machine-learning algorithm classifies browsed URLs overnight, and saves each classified paper in a central paper store. Explicit feedback and browsed topics form the basis of the interest profile for each user. The Quickstep recommender system uses a multi-class approach, allowing a profile in terms of domain concepts (research paper topics) to be built. The multi-class classification is less accurate than other binary classification systems, but allows class specific feedback and the use of domain knowledge (via an “is-a” hierarchy) to enhance the profiling process. Each day a set of recommendations is computed, based on correlations between user interest profiles and classified paper topics. Any feedback offered on these recommendations is recorded when the user looks at them. Users can provide new examples of topics and correct paper classifications where wrong in order to improve the training set over time.

Foxtrot (Middleton, De Roure, & Shadbolt, 2002) enhances the Quickstep system by employing the novel idea of visualising user profiles to acquire direct profile feedback. Foxtrot is an evolution of the Quickstep system with the increased number of supported interfaces, providing a research paper search interface, profile visualisation and feedback facility and email notification support. A static research paper ontology with many more classes is used, along with increased dimensionality reduction to cope with the increase in classes and hence term dimensions. The profiler takes profile feedback into account allowing users control over their own profiles. Lastly, a more collaborative recommendation algorithm is employed, taking into account the profiles of other similar users when deciding what to recommend.

The reported accuracy of both systems is relatively low, however, in theory it is possible to improve the accuracy by expanding the ontology the systems rely on. Ontological user profiles allow inference to be employed, allowing interests to be discovered that were not directly observed in the user’s behaviour (Spyrou, et al., 2014). Expanding the ontology

to include more relationships between topics would allow much more powerful inference, and thus give a significant boost to profiling accuracy.

This section looked at the various techniques for searching the Web and provided examples of approaches that rely on statistics and semantics. The next section will give an overview to effective methods for searching information within an internally stored knowledge base.

2.4.3 Effective Intranet Search

To maximise the relevance of data being searched, a processing system should be able to understand all information that can be automatically processed. A recently suggested approach to overcome this information processing problem is "concept search". Concept search (Jakus, et al., 2013) is an automated information retrieval method that searches unstructured text for information that has the same meaning as the information held within the search query. In other words, the ideas expressed in the retrieved information are relevant to the ideas in the text of the search query.

Various methods have been developed to retrieve and assemble data into information sets; from query languages to knowledge management tools. For instance, Autonomy (Autonomy, 2009) uses mathematical and statistical models that can handle both structured and unstructured data to form meaningful information to determine the categories of documents. This technology is used to identify patterns that naturally occur in text. Autonomy uses a semantic index that can restrict the scope of a concept to a sentence or paragraph in order to get a more accurate answer. Smartlogic (Smartlogic, 2001) employs ontologies that use meanings of the words and natural language rules to form information, and offer a common language for sharing knowledge in any given domain. Theory of Strongly Semantic Information (Floridi, 2004) looks at the completeness aspect of information from a different perspective and focuses on calculating semantic content.

2.4.3.1 Statistical methods – Meaning Based Computing

Systems applying statistics count how often groups of terms appear together within a document. This approach is based on the idea that if words occur together in similar contexts, they have similar meanings.

Meaning-based computing (Autonomy, 2008) is based on mathematical and statistical methods that can handle both structured and unstructured data to form meaningful information. This technology is used to identify the patterns that naturally occur in text. It can be achieved by frequency analysis of terms that correspond to specific concepts. By studying the predominance of one pattern over another, it is possible to calculate the probability that the content in question deals with a specific subject.

The challenge for the modern enterprise is trying to understand and extract the value that lies within this vast sea of data. When dealing with the challenge of unstructured information, it is clear, that plain search does not give desirable results. To maximise the relevance of information being searched, the processing system should be able to understand all information that can be automatically processed. This provides users with the ability to handle and maximise the value of this rich resource. Meaning Based Computing addresses the full range of information challenges and consequently forms the central requirement of major enterprise deployments all over the world.

The data sorting company Autonomy solves a problem that many businesses face – making sense of the masses of data and information in their enterprise systems. It is now a world leading company specialising in infrastructure software for the enterprise (Ashton, 2009) and which is spearheading the meaning-based computing movement. It works with an enterprise knowledge base and provides an automatic clustering of its content, avoiding silos and bringing the advantages of “horizontal search”.

Autonomy’s technology allows processing systems to harness the full richness of human information, forming a conceptual and contextual understanding of any piece of electronic data including unstructured information. This conceptual approach is unique to Autonomy (Vargas, 2008).The company’s core technology IDOL (Intelligent Data Operating Layer) (Autonomy, 2008), (Autonomy, 2009), provides a platform for the automatic categorisation, hyper-linking, retrieval and profiling of unstructured information, thereby enabling the automatic delivery of large volumes of personalised information.

The core engine is based on the mathematical works of Bayes and Shannon. This technology is used to identify the patterns that naturally occur in text, voice or video files. It can be achieved by frequency analysis of terms that correspond to specific concepts. By studying the preponderance of one pattern over another, IDOL understands that there is x% probability that the content in question deals with a specific subject. After that, the content’s digital essence is extracted and the unique signature of the concepts is encoded. This enables a host of operations to be automatically performed on emails, phone conversations, video,

documents and even people's interests. For better mathematical understanding of how it works, there is a need to Bayesian theory and Shannon's information theory. Bayes work centred on calculating the probabilistic relationships between multiple variables and when new information is obtained, determining the extent to which these relationships are affected.

Inferences are made taking into account all available information and answering the question of interest given the particular data set. Firstly, conditional probability (Gut, 2013) of text patterns is calculated:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.2}$$

Where P(A) and P(B) are the probabilities of text units (words, phrases) within the enterprise knowledge base. P(A | B) is the conditional probability of a text unit A co-occurring with a text unit B. A good example of this theory at work is Autonomy's agent profile technology (Autonomy, 2009). Users can create agents to automatically track the latest information related to their interests, and IDOL determines the relevance of a document based on the model of the agent.

Shannon's information theory (Ash, 1965), (Yeung, 2008) forms the mathematical foundation for all digital communications systems. The mathematical theory of information is based on probability theory and statistics, and measures information with several quantities of information. One of the most important is entropy (Roulston, 1999). Knowing the relationships between many variables allows the software to reveal the context of a piece of unstructured information. Having understood the meaning, Shannon's theory (Shannon, 1948) is then applied. The entropy of a discrete message space *M* is a measure of the amount of uncertainty one has about which message will be chosen. It is defined as the average self-information of a message *m* from that message space:

$$H(x) = -\sum_{i=1}^{n} p(i) \ln p(i) \tag{2.3}$$

Where ***x*** is a document from the enterprise knowledge base, *p(i)* is the relative probability of the co-occurring text patterns. The less frequently a unit of communication (for example a word or phrase) occurs, the more information it conveys. Thus ideas, which are rarer within the context of communication, tend to be more indicative of its meaning. By

studying the preponderance of one pattern over another, Autonomy's technology estimates the probability that the content in question deals with a specific subject.

This approach provides language independence of the text and allows the main concepts to be identified and prioritised. However, it captures only small part of the semantic information contained in a text leading to the wrong interpretation of meaning of content, which may undermine decisions and possible business operations. Natural languages contain a high degree of redundancy or nonessential content, information theory provides a framework for extracting the concepts from this redundancy.

2.4.3.2 Semantic Search

Semantic search provides a variety of techniques based on Artificial Intelligence and Natural Language Processing (NLP), and most of them rely on the use of auxiliary structures such as controlled vocabularies and ontologies to classify and find information. Ontologies (Gruber, 1993) provide shared representations of the entities and relationships characterising a domain, into which vocabularies of different systems can be mapped so to provide interoperability among them. Ontologies use meanings of the words and natural language rules to form information, and offer a common language for sharing knowledge in any given domain. The use of Semantic Web technologies (Fensel, 2005) allows publishers to expose their meta-data, thus, make the Web more easily searchable. Semantic search facilities are still developing, but practical implementations already exist and dramatically improve search results. Over the last couple decades, additional auxiliary structures of general interest have been constructed; data models that represent sets of concepts within a specific domain have been created in order to incorporate the relationships among terms.

In order to understand what a user is searching for, word sense disambiguation is used (Stevenson & Wilks, 2003). When a term can have several meanings, the disambiguation process takes into account other information presented in the text. All the knowledge used by the system is represented in the form of a semantic network, organised on a conceptual basis.

In semantic network every lexical concept corresponds to a node and is linked to other nodes by specific semantic relationships in a hierarchical structure. That makes it possible for each concept to be connected to the characteristics and meaning of the nearby nodes. Controlled vocabularies provide best results when search topics are narrowly defined with a standardised terminology. However, as language changes rapidly, they require

extensive human support for updates. Growing volumes of unstructured text bring unlimited new topics that need to be described and constantly introduced. Controlled vocabularies are also difficult to modify if concepts in a certain topic area change.

In the enterprise search world ontologies are widely used to support query answering against distributed and/or heterogeneous data sources (Stoilos, Grau, & Horroks, 2010). Ontologies and typical queries are often fixed at application design time. The Web Ontology Language OWL (The World Wide Web Consortium (W3C), 2009) is used to access ontology-based data. Ontology provides the vocabulary used to formulate queries, and a conceptual model that is used in computing query answers. In a semantic Web setting a typical scenario would involve the use of OWL ontology to answer queries over datasets: ontology is used to describe the meaning of the data stored in various sources, and query answers reflect both the data and the knowledge captured in the ontology.

Efficient management and querying of large amounts of data is a core problem for a growing range of applications in fields as diverse as business and defence. In order to facilitate interoperability, such applications often use standard data models and query languages. In particular, Resource Description Framework (RDF) (RDF Working Group, 2004) provides a standard model for semi-structured data, SPARQL (Prud'hommeaux & Seaborne, 2008) is a standard query language for RDF, and ontology languages such as OWL (Horrocks, Patel-Schneider, & van Harmelen, 2003) and OWL 2 (Grau B. H., Motik, Parsia, Patel-Schneider, & Sattler, 2008) can be used to describe background knowledge about the application domain. Thus, answering SPARQL queries over RDF data sets structured using an OWL ontology is a key service in ontology-based information systems.

Some systems are guaranteed to compute all answers for each query, ontology, and data set (Grau, Motik, Stoilos, & Horroks, 2012). Completeness, however, comes at the cost of scalability, as answering queries over OWL 2 ontologies is of high computational complexity. Thus, complete systems often fail to meet the scalability demands of applications that manage data sets consisting of hundreds of millions or even billions of assertions. Scalability of query answering can be ensured by restricting the expressive power of the ontology language to the level that makes provably complete reasoning tractable (Zhou, et al., 2012).

However, even within a restricted environment, scalability remains one of the biggest issues for ontology-based search systems. When using an expressive ontology language such as OWL, computing query answers can be very costly, and in a semantic Web setting, datasets may be extremely large. For ontology-based applications in order to improve

scalability, many semantic Web query answering systems give up completeness (Stoilos, Grau, & Horrocks, 2010). In other words they do not guarantee to return all query answers.

The authors (Grau, Motik, Stoilos, & Horroks, 2012) argue that in order to achieve scalability of query answering, the developers of Semantic Web applications are often forced to use incomplete OWL 2 reasoner, which fail to derive all answers for at least one query, ontology, and data set. The lack of completeness guarantees, however, may be unacceptable for applications in areas such as health care and defence, where missing answers can adversely affect the application's functionality. Furthermore, even if an application can tolerate some level of incompleteness, it is often advantageous to estimate how many and what kind of answers are being lost.

Thus, in the Semantic Web applications, completeness is often not strictly required (Stoilos, Grau, & Horrocks, 2010). Consequently, many systems have chosen to give up completeness. The trade-off between completeness and efficiency is obviously a continuum. Incomplete systems find some, but not all the answers implied by the ontology and the data. A major difficulty with incomplete conjunctive queries answering systems, with the ontology providing the vocabulary used in query, is determining how incomplete they are.

In (Grau, Motik, Stoilos, & Horroks, 2012) a possible solution to check the completeness of an ontology reasoner was proposed; the authors present a novel logic-based framework that allows one to check whether a reasoner is complete for a given query and ontology. The proposed technique allows checking whether the reasoner is guaranteed to compute all answers to the query with respect to the ontology and an arbitrary data set. Since ontologies and typical queries are often fixed at application design time, their approach allows application developers to check whether a reasoner known to be incomplete in general is actually complete for the kinds of input relevant for the application.

The authors (Grau, Motik, Stoilos, & Horroks, 2012) also present a theoretical and practical foundation for the design of future ontology-based information systems that maximise scalability while minimising or even eliminating incompleteness of query answers. The proposed technique can be used for comparing the effectiveness of two given reasoners for a given query and ontology over a given data set. This allows application developers to select the reasoner that provides the highest degree of completeness for the query and the ontology that is compatible with the application's scalability requirements.

The reasoner is incomplete if there exists at least one query, ontology, and data set for which the reasoner does not return all answers to the query (Grau & Stoilos, 2011). However, incomplete reasoners can handle large data sets and they often provide the best

practical choice for developers of ontology-based applications. In order to verify that the selected reasoner meets the application's requirements, developers typically resort to empirical testing, in which they check the reasoner's answers w.r.t. the application ontology and queries for representative data sets.

As analysed in (Grau, Motik, Stoilos, & Horroks, 2012) empirical completeness testing, however, has several important limitations. First, tests are not generic, as data sets used for testing typically have a fixed and/or repetitive structure, which can skew test results. Second, test data is not exhaustive, as completeness is tested only w.r.t. a limited number of data sets. Finally, query answers may not be verifiable: since complete reasoners fail to handle large data sets, they often cannot compute the control answers needed to check the answers produced by an incomplete reasoner.

2.4.3.3 Theory of strongly semantic information

Theory of strongly semantic information (Floridi, 2004) looks at the completeness aspect of information and focuses on calculating semantic content.

Shannon's information theory (Shannon, 1948) is widely known as a technical approach that involves quantity analysis of information (note, that here the terms "information" and "data" are used interchangeably). A widely used (in mathematical theory of communication) probability approach to the quantity of information is based on calculating the information content of a message as the amount of uncertainty reduced when the message is received. It is associated with the elimination of possibilities represented by that message.

Italian philosopher Luciano Floridi from University of Oxford is exploring the area of philosophy of information in his work (Floridi, 2004). He considers information as knowledge that is subject independent, and calls it "semantic information". His quantitative theory of strongly semantic information tends to be possible on the basis of a calculus based on truth-values and degrees of discrepancies with respect to a given situation, rather than probability distributions. The main hypothesis supported has been that semantic information encapsulates truth, and hence that false information fails to qualify as information at all.

Floridi's approach is based on truth values rather than probability distributions. Probability analysis gives paradox results, when applied to information that holds false semantic content, or is a false statement. If Shannon's information theory is applied, the result leads to the paradox – if the statement becomes less likely, it gradually increases the

informational content of it. Thus, false statement contains most of information and this is semantically wrong.

In (Floridi, 2013) the author describes the idea of the theory as to define semantic-factual information in terms of data space, as well-formed, meaningful and truthful data. In other words, the basic idea is that the more accurately a statement corresponds to the way things actually are, the more informative it is. There are two extremes. First, if a statement is true and is a tautology (is true in every possible interpretation and every situation supports the statement. Example: "The sky is blue or it is not the case that the sky is not blue"), it yields no information. Second, if a statement is false and is a contradiction (no possible situation supports the statement. Example: "It's raining and it's not raining."), it also yields no information. Between these two points there are contingently true and contingently false statements with varying degrees of vacuity and inaccuracy.

In his work (Floridi, 2004), Floridi gives “general definition of information” in terms of data and meaning as follows. $\boldsymbol{\sigma}$ is an instance of information, understood as semantic content, iff:

1) $\boldsymbol{\sigma}$ consists of one or more data (the stuff of which information is made);
2) the data in $\boldsymbol{\sigma}$ are well formed (clustered together correctly, according to the rules (*syntax*) in the chosen system, code or language);
3) the well-formed data in $\boldsymbol{\sigma}$ are meaningful (must comply with the meanings (*semantics*) of the chosen system, code or language).

Theory of strongly semantic information constrains the probabilistic approach by requiring first a qualification of the content as truthful. Once the content is so qualified, the quantity of semantic information in $\boldsymbol{\sigma}$ is calculated in terms of distance of $\boldsymbol{\sigma}$ from the situation/resource $\boldsymbol{w}$ that $\boldsymbol{\sigma}$ is supposed to model. Total distance is equivalent to a $\boldsymbol{\sigma}$ true in all cases (all possible worlds or probability 1), including $\boldsymbol{w}$ and hence minimally informative, whereas maximum closeness is equivalent to the precise modelling of $\boldsymbol{w}$ at the agreed level of abstraction.

Let $\boldsymbol{\theta}$ refer to the distance between a true $\boldsymbol{\sigma}$ and $\boldsymbol{w}$ and can be read as “the degree of support” offered by $\boldsymbol{w}$ to $\boldsymbol{\sigma}$. The formula to calculate the *degree of informativeness* $\boldsymbol{\iota(\sigma)}$ in relation to the distance $\boldsymbol{\theta(\sigma)}$:

$$\iota(\sigma) = 1 - \theta(\sigma)^2 \tag{2.4}$$

Figure 2.11 below shows the graph generated by the equation and includes both positive and negative values of distance. Distance θ ranges from -1 (contradiction) to +1 (tautology), meaning that the shorter the distance from the 0 point, the closer the statement to the truth.

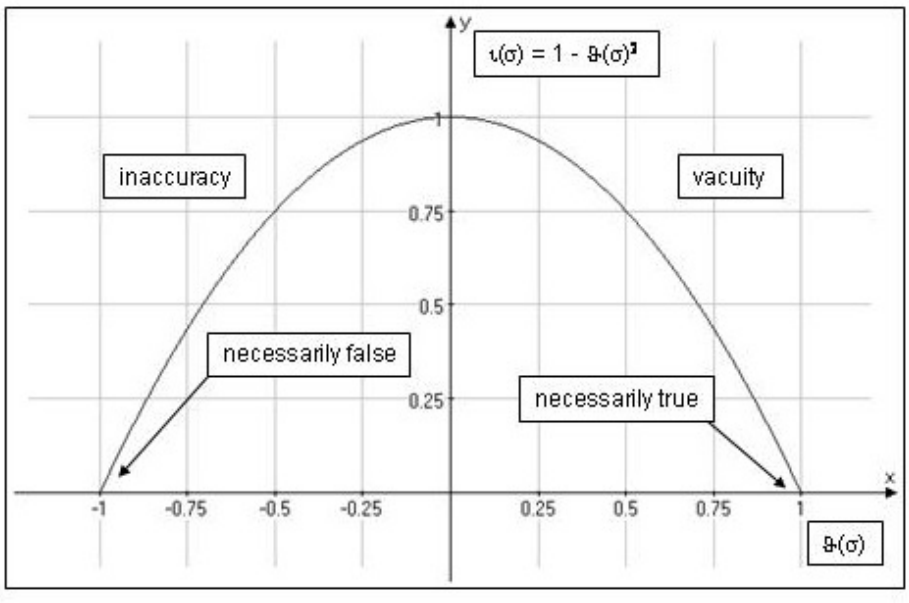


Figure 2.11 – Degree of informativeness ι(σ) in relation to the distance θ(σ)

If an instance of information σ has a very high degree of informativeness ι (very low distance θ), then it contains a large quantity of semantic information and vice versa. In the graph, the quantity of semantic information contained in σ relative to $\iota(\sigma)$ is the area under the curve – the definite integral of the function $\iota(\sigma)$ on the interval [0, 1].

T*he* maximum quantity of semantic information α (the shaded area Figure 2.12) is carried by an instance of information which is fully accurate, precise and contingent truth, and whose $\theta(\sigma) = 0$. Hence:

$$\int_0^1 \tau(\sigma)dx = \alpha \tag{2.5}$$

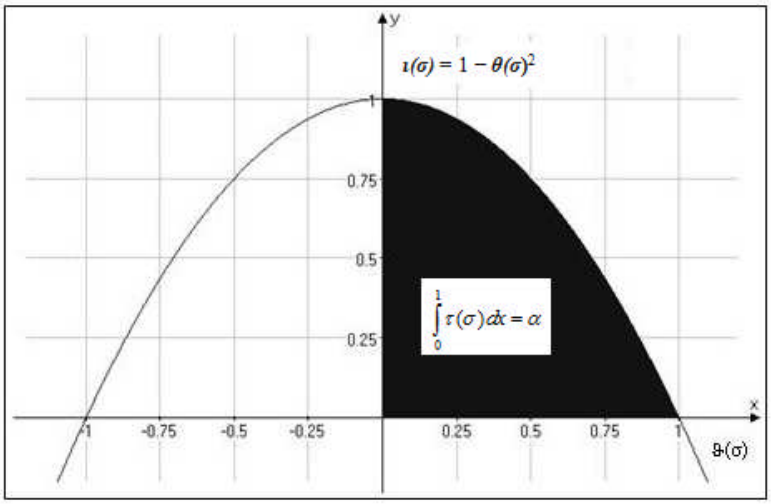


Figure 2.12 – Maximum quantity of semantic information

For an instance of information $\boldsymbol{\sigma}$ that contains some information that is simply irrelevant or redundant (i.e. "informational waste") the amount of vacuous information $\boldsymbol{\beta}$ is also a function of the distance $\boldsymbol{\theta}$ from $\boldsymbol{w}$:

$$\int_0^{\theta} \tau(\sigma)dx = \beta \tag{2.6}$$

Shaded area in Figure 2.13 below shows the amount of vacuous information $\boldsymbol{\beta}$.

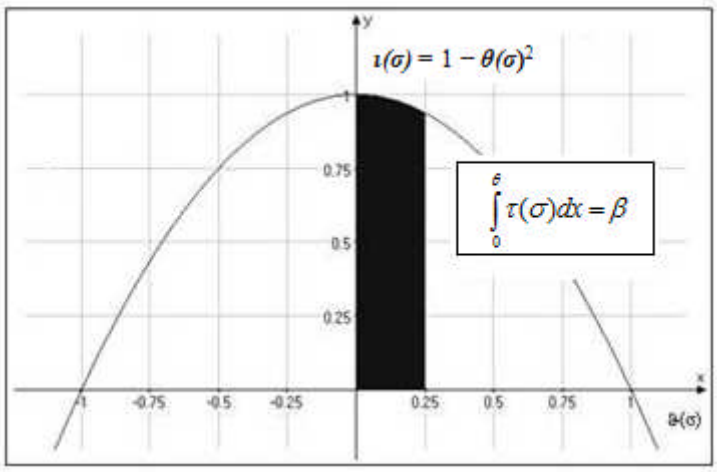


Figure 2.13 – Amount of vacuous information $\boldsymbol{\beta}$ in $\boldsymbol{\sigma}$

Clearly, the amount of semantic information in is simply the difference between $\boldsymbol{\alpha}$ (the maximum amount of information that can be carried in principle by $\boldsymbol{\sigma}$) and $\boldsymbol{\beta}$ (the amount of vacuous information actually carried by $\boldsymbol{\sigma}$):

$$\gamma = \alpha - \beta \tag{2.7}$$

Shaded area in Figure 2.14 shows the amount of semantic information in $\boldsymbol{\sigma}$.

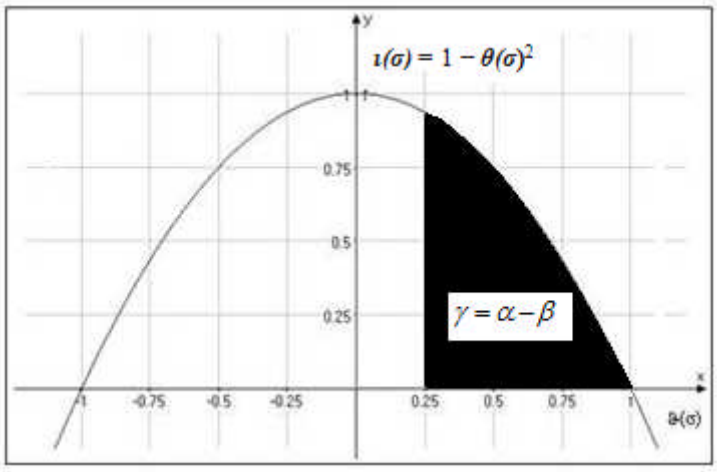


Figure 2.14 – Amount of semantic information $\boldsymbol{\gamma}$ in $\boldsymbol{\sigma}$.

In contrast to the probability approach to the quantity of information, the theory of strongly semantic information overcomes what is known as Barr-Hillel-Carnap paradox (Carnap & Bar-Hillel, 1952), which states that a false sentence is highly informative since there is no value in whether information it carries is true or false. Theory of strongly semantic information requires first a qualification of the content as truthful. Once the content is so qualified, the quantity of semantic information in $\boldsymbol{\sigma}$ is calculated in terms of distance of $\boldsymbol{\sigma}$ from the situation/resource $\boldsymbol{w}$ that $\boldsymbol{\sigma}$ is supposed to model.

However, qualification of the content as truthful means identification and qualification of all possible states that correspond to the given situation. This makes Floridi's approach too theoretical to be applicable to the real world situations.

2.5 Issues with Web-based Intelligence Gathering

Decision making is a fundamental process of management. The success of a business largely depends on the effectiveness of the decisions made. During the process of decision making one can face uncertainty, unpredictability, or probabilistic nature of the result that is influenced by a number of factors, both internal and external. The quality of decisions made in business and government correlates directly to the quality of the information used to formulate the decision (Eppler, 2006). Most of the information used for intelligence analysis will, in the future, be harvested from the Web as this is becoming the richest source. Intranet held information (company's knowledge base) can be efficiently manipulated by enterprise search systems based upon either semantics such as ontologies, or meaning-based computing. These technologies imply comprehensive (and often automatic) indexing and tagging of the Intranet knowledge base textual information. Existing Web, as originally described in (Berners-Lee, 2001), was expected to evaluate into Semantic Web, that encourages simply the inclusion of semantic content in Web pages, making it not only human readable, but also machine readable. However, most of the current Web remains poorly semantically tagged, making it impossible to apply effective enterprise search methods to Web-based intelligence information extraction. If the Web is to be used for improving decision making, then new more effective search methods must be developed in order to collect and correlate the best information. This new search method may be used to harvest Web data in accordance with carefully controlled parameters and transferred to the Intranet knowledge base where upon enterprise search technologies may be then applied in the usual way.

It should also be noted that an Intranet knowledge base can become too historic and Web-based knowledge more effectively reflects the current state of the world. Regular updates to an Intranet knowledge base would make sense but requires time and resources.

Previous sections provided overview to different existing systems and methodologies used for information discovery from the Web and Intranet. However, none of the available search methods (i.e. Web-search engines, enterprise-search solutions) can support Web-wide collection of information that is relevant to the search topic, providing its high quality and completeness. This is due to various aspects of the search environment, such as data infrastructure, content organisation, indexing, or ranking of search results. More importantly, these issues affect the decisions based on the information of poor quality.

According to (Schum, 2001), all evidence has three major credentials: relevance, credibility, and inferential force or weight. Relevance concerns the degree to which the evidence bears upon the hypothesis under consideration. Credibility means the degree to which the evidence is believable. Inferential force concerns the strength of the relationship between evidence and hypothesis – the degree to which the evidence sways our belief in the hypothesis. Approaches for dealing with the weight or strength of evidence include both qualitative and quantitative aspects of the reasoning process adopted to draw inferences from it (e.g. probability theory, logical reasoning, etc).

The initial objective of this research was to research and develop a quality measure that can be assigned to an organisation's decisions that are based in part or whole on the information that has been derived from the Internet. However, after exploring the problem and existing solutions further, it was concluded that available approaches lack one or more of the fundamental metrics for Web-based information extraction: *relevance, completeness and quality*.

Effective computerised support for decision makers and intelligence analysts must support processes of evidential reasoning. This support, as noted by (Laskey, Schum, Costa, & Janssen, 2008), requires the ability to represent, store, and manipulate evidence, hypotheses, and arguments, relating evidence to hypotheses. Such representations must be stored in a computational structure. Such representation can be constructed using the languages and tools commonly applied in the discipline of ontological engineering.

Traditional search engines (Google, Bing, etc.) have proven their effectiveness in finding on-line stored information, as well as some of them provide search through locally stored documents. The performance of a traditional Web-search engine is based on finding the keywords from its index base that match with a search query string. This approach is fast, but does not allow finding the documents that share the same idea with the query, that is expressed in different words. Another issue associated with traditional Web-search engines is that they present their results ordered by pseudo-relevance, such as dependant on the popularity of a Web-page. That is why Web-pages with relevant and even high quality content can be found at the bottom of the search list, or even become overlooked.

Enterprise-search engines (Autonomy, Smartlogic, etc.) are effective when searching for relevant documents within locally stored knowledge base - Intranet. Such systems are based upon semantics or statistics, and these technologies require complex processing of large amounts of text. However, enterprise-search systems cannot be currently applied to Web-wide collection of information due to various Web-related issues such as lack of

semantic tagging of Web-content. Effective Intranet search methods are just too complex for Web application. The finite size of document base makes it possible to perform concept search to find documents that have the same idea within the context. But there are various issues that do not allow concept processing of the Web content. First of all, the Web content is contained mainly in silos, and extremely badly tagged in comparison to the level of tagging in enterprise knowledge base. More than that, wrong interpretation of meaning of content may undermine decisions and possible business operations. But the major issue is the amount of information that needs to be processed. It is just technically impossible to process huge Web-content and apply the algorithm within a reasonable time period.

Methods reviewed in this chapter allow estimating relevance and completeness of the search results. But there is currently no metric for measuring the quality of the gathered information and know how good it is. Selection of data for a decision information gathering task can be complicated and dangerous (lead to the wrong decision making), as there is no obvious way of measuring the quality of the information gathered.

In information world and one where internationalism is becoming the norm, decision makers will rely increasingly on the Web to augment the organisation's database and human intelligence for both strategic and tactical decisions. For data from the Internet to be used in these important circumstances the quality of data harvested from the Internet needs to have an associated quality measure. This quality measure may then be used to assign a quality mark against the decision made, thus allowing an organisation to judge the integrity of the decision made.

Hitherto only limited research has been undertaken regarding the quality of decisions, where the Web provides a significant element of the supporting information. For example, authors of (Agichtein, Castillo, Donato, A., & Gilad, 2008) discuss quality of user-generated content. It suggests a comprehensive graph-based model of contributor relationships and combined it with content- and usage-based features.

The study in (Kopcke, Thor, & Rahm, 2010) addresses the record linkage problem. Determining whether two different records represent the same real world object is much harder in the context of data on the Internet. This article adopts a machine learning technique to semi-automatically determine suitable match strategies with a limited amount of manual effort of training.

(Gao & Houben, 2010) present an approach that discusses two specific tasks: the creation of a representation model of trust in Web data for specific data consumers, and the

support for three strategies to assess the trustworthiness of data that they consume: vocabulary-based , triple-based and interlinking-based strategies.

(Capiello, Daniel, Matera, & Pautasso, 2010) focuses on assessing the quality of mashup applications. The research aims to assess the quality of the information a mashup provides, which requires understanding how the mashup has been developed, what its components look like, and how quality propagates from basic components to the final mashup application.

Raiber & KURLand (2013) addressed the task of query-performance prediction for Web search. They devised a query-performance prediction approach that utilises query-independent document quality measures. These measures include the probability that the document is not spam as determined by a spam classifier; its PageRank score (Gomes & Smith, 2000); and estimates of the “richness” of the language used in the page.

A pertinent work on measuring quality via entropy is presented in (Kao, Lin, Ho, & Chen, 2002), in which the authors correlate quality of new information with the measured entropy value of the text on the Web page. The idea of the approach is that terms appearing in fewer pages carry more information to the user. Terms are extracted from anchor texts.

Shannon's entropy is used to represent the information strength of terms. It is applied on the term-document matrix to calculate the entropy, which is expressed as:

$$-\sum_{i=1}^{n} p_i \log p_i, \tag{2.8}$$

where p_i is the probability of event ***i*** and ***n*** is number of events.

By normalising the weight of a term to be from 0 to1, the entropy of term *Ti* is:

$$E(T_i) = -\sum_{j=1}^{n} w_{ij} \log w_{ij}, \tag{2.9}$$

in which $\boldsymbol{w_{ij}}$ is the value of normalised term-frequency, $\boldsymbol{w_{ij}}$ is an entity in the term-document matrix and represents the weight of a term in a page:

$$w_{ij} = \frac{tf_{ij}}{\sum_{k=1}^{n} tf_{ik}}, \tag{2.10}$$

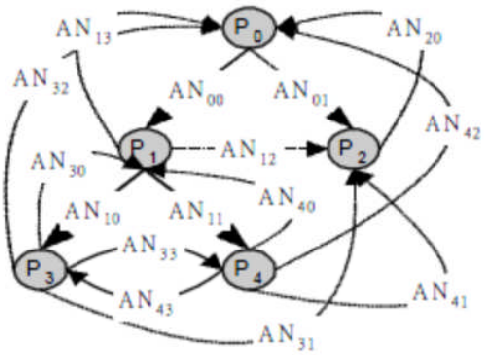
where $\boldsymbol{tf_{ij}}$ is the term frequency of term *i* in page *j*.

The entropy value of a term is then normalised to the range [0, 1], the base of the logarithm is chosen to be the number of pages. $\mathbf{E(T_i)}$ then becomes:

$$E(T_i) = -\sum_{i=1}^{n} w_{ij} \log_n w_{ij}, \tag{2.11}$$

where ***n*** = **|D|**, D is a set of pages.

Taking the authors' example (figure 2.15), let a sample Web-site contain five pages [P0, P4] with the corresponding adjacent matrix showing the structure of the links between the pages.



	P0	P1	P2	P3	P4
P0		1	1	0	0
P1	1		0	1	1
P2	1	0		0	0
P3	1	1	1		0
P4	1	1	1	0	

Adjacent Matrix

Figure 2.15 – Example structure of the links between the pages

Then the entropy of anchor AN_i is defined as an average entropy of all terms in AN_i: where T_1, T_2, .. , T_k are terms in anchor AN_i:

$$E(\mathrm{AN}_i) = \frac{\sum_{j=1}^{k} E(T_j)}{k}. \tag{2.12}$$

Terms that have their entropy values close to "1" provide very little information for users. In general, these terms come from links of advertisement banners and navigation panels. Hence, anchors that contain high entropy terms less informative. And users can find information relevant to the terms that can only be found through anchors with entropy 0. Users are usually more interested in such information of smaller entropy. This means that an anchor with a smaller entropy term should be assigned a larger weight than the one with a larger entropy term.

The goal of information discovery is to reduce the uncertainty by increasing acquired knowledge. Recent research projects aimed at discovery of UUs focus primarily on ontology-based knowledge acquisition techniques. (Lehmann, Schüppel, & Auer, 2007) presented a new user interface 'DBpedia' to explore a large ontology-based data set by finding connections between different objects, thus, discovering UUs. The core of DBpedia is

in the form of an ontology that represents background knowledge comprising structured information extracted from Wikipedia. However, this solution is limited to searches within the DBpedia data set only.

Further work in the discovery of UUs was undertaken by the TORISHIKI-KAI project (Torisawa, et al., 2010) which displays all relevant keywords extracted from a large Web-archive as having a particular semantic relationship to the search topic. Even though TORISHIKI-KAI summarises a large volume of Web documents, the search system is designed to suggest relevant UUs under just three semantic categories: troubles, methods and tools.

Another noteworthy method is the 'Pattern-based Understanding and Learning System' (PULS) which is designed to extract news information from several domains (Huutunen, Vihavainen, Du, & Yangarber, 2012). Its Web-crawler finds relevant articles using a keyword-based Web-search. The rule-based information extraction module analyses the plain text from the news feed and transforms this text into database records. Its declassifier determines the relevance of the selected events to a particular use.

This section provided overview to issues associated with current techniques that are applicable to information gathering form the Web as well as from an internally held knowledge base. All of the above methods have limitations in terms of information sources or functionality. There is a need for solution that is open to perform search and analysis through any Web-page available to the search engine. Modification of existing theories can enable the information content to be tested and allow harvesting Web-wide data in accordance with controlled parameters in order to subsequently transfer this data to a knowledge base where enterprise search technologies may be applied in the traditional way.

In addition, it is important to keep track of the information growth rate due to the risk of information level to become critical and exceed the optimal level, so that it creates the information overload and may lead to the lower decision quality. Therefore, the aim for the analysts will be to collect as much relevant information as possible, thus not exceeding the optimal amount of it and, hence, keeping the decision quality as high as possible. In order to overcome the above issues, a new solution has been developed to systematically identify the semantic relationship between the search topic and the Web-content, and be able to provide information analysts with a quality measure to judge the importance of found information.

The new algorithm proposed in this research combines the traditional Web-search approach with semantic analysis in order to overcome the issues associated with both types of search engines and significantly enhances the collected information from a richness

viewpoint. The proposed algorithm goes beyond the traditional query search as far as it identifies topic-related information from the Web using methods from Natural Language Processing (NLP), as well as sorts the relevant documents by quality of the information inside, and not by the popularity measure in contrast to traditional search engine.

In summary, the proposed approach relies on evidence-based NLP which may be considered as comprising three integrated processes that are as a whole iterative. Firstly, the application of NLP methods to enable the filtering of Web-search results to form a set of relevant information, thus overcoming the search engine keyword and ranking mechanisms that limit the use of a search engine approach. Thus, the captured sets of KK, KU and UU are semantically related and, therefore, relevant to the topic being considered. Secondly, this captured set is subjected to the application of Grounded Theory where UU are specifically identified and used to test the completeness of the evidence. Thirdly, the application of the Evidential Analysis is used to test the quality of gathered information and hence setting a quality parameter for the efficacy of the eventual decision-making process. The three processes together are applied iteratively to the Web with an expanding query base using converted UU in order to identify the best information for the target decision process.

Chapters 3 and 4 will provide details of the proposed approach.

2.6 Summary

This chapter introduced the details of the problem of imperfection of the existing methods that may be used to support Web-wide collection of information that is relevant to the search topic, providing its high quality and completeness. Various current methods for information gathering from the Web and locally stored sources have been analysed and the associated issues described. Among them there were methods for statistical and semantic search of relevant information applied to both the Web and the Intranet.

- Latent Semantic Indexing – Web-search statistical approach. It is a powerful tool that extracts the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts. Its full power cannot be achieved when used with a short query. When applied to the locally stored data collection, a full index needs to be built before searching, which is not efficient with dynamically changing data sources and average machine capability.

- Google PageRank – sorting algorithm. It places popular Web-pages towards the top of the search result list. "A page is important if it is pointed to by other important pages". It is query independent and doesn't give the results that are most relevant to the search idea, but that enough people consider worth linking to.
- Ontologies in recommender systems – semantic Web-search approach. It creates user profiles from unobtrusively monitored behaviour and relevance feedback. A collaborative recommendation algorithm can be employed, taking into account the profiles of other similar users when deciding what to recommend. The solution is, however, limited to the ontology subject domain and cannot be used across the entire Web.
- Meaning-based computing – enterprise search statistical approach. This technology is used to identify patterns that naturally occur in text, voice or video files. It relies on calculating the probabilistic relationships between multiple variables and when new information is obtained, determining the extent to which these relationships are affected. This approach is not feasible when applied to a much larger set of the Web data.
- Ontology-based information systems – enterprise search semantic approach. It incorporates the relationships among terms by relying on created data models that represent sets of concepts within a specific domain. All the knowledge used by the system is represented in the form of a semantic network, organised on a conceptual basis. However, most of the current Web remains poorly semantically tagged, making it impossible to apply effective enterprise search methods to Web-based intelligence information extraction. In addition, as language changes rapidly, such systems require extensive human support for updates. Finally, computing all answers for each query, ontology, and data set comes at the cost of scalability, thus, not feasibly applicable to the Web.
- Theory of strongly semantic information – calculates the information content of a message as the amount of uncertainty reduced when the message is received, and is associated with the elimination of possibilities represented by that message. This method avoids paradox results produced by probability analysis, when applied to information that holds false semantic content, or is a false statement. The advantage of the method is that it is based on truth values rather than probability distributions – the more accurately a statement corresponds to the way

things actually are, the more informative it is. However, qualification of the content as truthful means identification and qualification of all possible states that correspond to the given situation. This makes the "theory of strongly semantic information" approach too theoretical to be applicable to the real world situations.

In addition to the issues associated with relevance and completeness, Chapter 2 looked at the limited available research that provides a metric for measuring the quality of decisions, based on a significant part of information from the Web.

- A graph-based model of author relationships combined with content- and usage-based features for quality estimation of user-generated content.
- A machine learning technique to determine whether two different records represent the same real world object.
- A representation model of trust in Web data for specific data consumers.
- A method for query-performance prediction for Web search that utilises query-independent document quality measures.
- Measuring quality of the data source via measuring entropy value of the text on the Web page.

This chapter also looked at the recent research projects that address the discovery of unknown unknowns.

- A user interface 'DBpedia' for exploring a large ontology-based data set by finding connections between different objects, thus, discovering UUs. The solution is limited to searches within the DBpedia data set only.
- TORISHIKI-KAI system displays all relevant keywords that have some semantic relationship to the search topic. This system, however, supports only three semantic categories: troubles, methods and tools.
- "Pattern-based Understanding and Learning System" extract news information from the selected news domains.

All of the above research tends to focus on specifically selected topics and have limited functionality.

Thus, there is a need for a new solution that, in contrast to the above methods, has no limitations for Web-sources and is open to perform search and analysis through any Web-page available to the search engine. This new search method may be used to harvest Web data in accordance with carefully controlled parameters and transferred to the Intranet

knowledge base where upon enterprise search technologies may be then applied in the usual way.

Chapter 3

Details of the Proposed Approach

3.1 Overview to Chapter

This chapter will address the following research step:

Investigate research concepts and methods that can be used (or further developed to be used) with a conventional search engine to enrich information retrieval through the discovery of unknown unknowns. Such methods would provide:

- *an effective approach to formulate a topic-focused collection of the Web content that is semantically related to the search topic,*
- *a mechanism for tracking the growth of information and the reduction of uncertainty during information discovery,*
- *a quality measurement of the extracted content.*

The aim of this chapter is to introduce and develop the algorithm of the new Web search method to meet the objectives of this thesis. This chapter describes the logic behind the proposed method and will focus on both qualitative and quantitative details of each stage of the algorithm. The major part of this chapter will focus on the elements used in the main processing part of the algorithm. Among them are the existing methods from Natural Language Processing, Grounded Theory and evidential analysis – each of these individual methods will be described in detail in order to establish the efficacy of the overall method.

Chapter 2 provided an overview to currently available research methods. Semantic search methods have been shown to work effectively with enterprise knowledge, but not with the Internet. This is due to the numerous obstacles the technology faces, such as higher cost of the software and lack of computational power and storage capacity in comparison with traditional keyword search. Grounded theory has been successfully used in building a hypothesis using interviews, while evidential analysis is effective in investigating the

evidence associated with a hypothesis. But no single approach has been shown to effectively filter and test the gathered evidence using all sources of information.

The problem definition for this research states that none of the available search methods (i.e. Web-search engines, enterprise-search solutions) can support Web-wide collection of information that is relevant to the search topic, providing its high quality and completeness.

In contrast to the currently available research methods (identified in Chapter 2), which tend to focus on specifically selected topics and information sources, the solution presented in this thesis has no limitations for Web-sources and is open to perform search and analysis through any Web-page available to the search engine. This new search method may be used to harvest Web-wide data in accordance with controlled parameters and subsequently transfer this data to a knowledge base where enterprise search technologies may be applied in the traditional way.

The steps followed to develop the algorithm are: detailing the proposed logic for the research (section 3.2); a detailed analysis of the search problem (or information target) is undertaken as a preparation for the full search process (section 3.3); integration on the main elements that form the algorithm where the main mathematical theory used for the research is developed and explained in detail in section 3.4; the elements are brought together in section 3.5 to provide a visualisation of the algorithm. A detailed summary of the research approach in provided in section 3.6 in preparation for the next chapter.

3.2 The Logic of the Proposed Solution

3.2.1 Base discussion for the logic of the proposed solution

Recalling the objective of this research, it has been established to develop an algorithm that supports topic- related search for Web-based information. It will also estimate the quality of the extracted content, thus proving it is significantly enhanced compared to the traditional Web-search approaches.

In order to overcome the issues associated with the traditional Web-search approaches, there is a need to systematically identify the semantic relationship between the search topic and the Web-content, and be able to provide information analysts with a quality

measure to judge the importance of found information. Moreover, it is important to keep track of the information growth rate due to the risk of information overload becoming critical and exceeding the optimal level. An information overload may lead to lower decision quality. Therefore, the aim for the analysts will be to collect as much relevant information as possible, thus not exceeding the optimal amount and, hence, keeping the decision quality as high as possible without overload.

Combining known methods has the potential to reduce uncertainty through the medium of evidential analysis. A strategy has to be developed that can be used to automatically find and analyse the collected information in order to improve the decision making and decrease the level of uncertainty associated with it.

As Chapter 2 explained, there are several ways of extracting textual information from digital sources. All of them have inherent strengths, as well as limitations and cannot effectively support an individual in a Web-based intelligence information formulation. This thesis uses a hybrid approach – a combination of the traditional keyword search with semantic methods to collect the information from the richest source – the Web.

The new method proposed in this thesis supports Web-based search for intelligence information formulation based on the structured combination of methods from computational linguistics and evidential analysis, and other methods which may prove relevant.

3.2.2 Development of the logic for the proposed solution

Following from exploring of the problem and existing solutions in Chapter 2, it was concluded that currently available approaches lack one or more of the fundamental metrics for Web-based information extraction:

- relevance;
- completeness;
- quality.

The proposed algorithm is aimed to address each of these three aspects and thus overcome issues of the existing solutions described in Chapter 2.

It was decided to combine the benefits of traditional keyword search with semantic methods to collect the information from the Web that is semantically relevant to the search topic. In order to address the issue of semantic relevance and completeness, the search

algorithm will rely on a traditional search engine. Google was chosen for its ability to access a large part of the Web. Moreover, this search engine is not limited to a selected range of Web-sources, in contrast to some of the existing sematic search solutions. The use of LSI will allow getting large amount of responses fast; however, the resulting search list will contain many false positives due to the keyword search picking up content where the selected search terms are present, but not necessarily express the ideas similar to those in the search topic.

The search topic needs to be defined in textual format to provide enough context for application of a semantic method that will compare ideas behind the search topic and content derived from the Web. This means, that there will be a need in an effective method of extracting of the main content of a Web page in order to benefit from application of a semantic method to calculate semantic closeness score as quantitative measure for assessment. This approach will overcome limitations associated with the traditional keyword-based search as it will filter out the irrelevant Web-pages (which are only mechanically responsive to the search terms) and leave those results that share similar ideas with the search topic. At the same time, this approach will provide a scalable solution that is flexible enough to adapt to the dynamically changing Web environment and handle the potentially large amount of information to process.

The search topic expresses the KKs and will be defined as an initial knowledge base, which is treated as a benchmark for semantic relatedness of the content. In addition to the initial knowledge base, search objectives need to be defined – they will reflect the direction of the search and represent the KUs on the search topic.

In order for the search method to comply with Ashby's Law of Requisite Variety, the completeness aspect needs to be addressed. According to the Law, the variety of the system must at least match the variety of the environment it controls. As such, relevant Web-pages will be added to the existing knowledge base, thus, enriching the available knowledge on the topic and enhancing the basis for the effective decision making. It is important to keep the size of the knowledge base within the boundaries of the necessary level of completeness in order to avoid the information overload or deficit, thus, supporting the decision quality at its highest. Thus, there is a need for a parameter that would reflect the growth rate for the size of the knowledge base and indicate when the search topic is close to exhaustion, or needs further expansion. In other words, this parameter would assess the developing knowledge base from the richness viewpoint. It will show whether there is a need for further Web-search to discover more unknown unknowns, or that further searching will not add any valuable information that will enhance the effectiveness of the decision making.

The quality of the discovered information also needs to be addressed. As covered in Chapter 2, existing methods rely on limited selected sets of information sources. The proposed solution already overcomes this limitation as it is aimed at searching across all data available for a traditional Web-search engine. However, as stated in the research objective, high quality information will need to form a topic-focused knowledge base. Therefore, there is a need to develop a quality metric that relies on semantic parameters and assesses the quality of new information the semantic viewpoint.

In order to accommodate the large volumes of information to process, the search and filtering process will need to be iterative. If a single search query corresponds to a direction of the search, processing of its results will correspond to a single iteration. Similarly, within a single "search query" iteration, each Web-page needs to be processed to get its extracted text and run assessment for its semantic, completeness and quality scores.

3.2.3 Logic for the new algorithm

The following diagram at Figure 3.1 presents a schematic overview of the steps involved. The proposed method consists of the following 7 steps:

1. Identify target knowledge and search objectives;
2. Perform Web-search;
3. Pre-process Web-pages;
4. Perform semantic distance measurement;
5. Build/enhance evidence;
6. Test the evidence quality;
7. Identify further search direction, if necessary. Here it may be necessary to expand or refined the query; this represents an iterative process until a calculated level of quality has been reach.

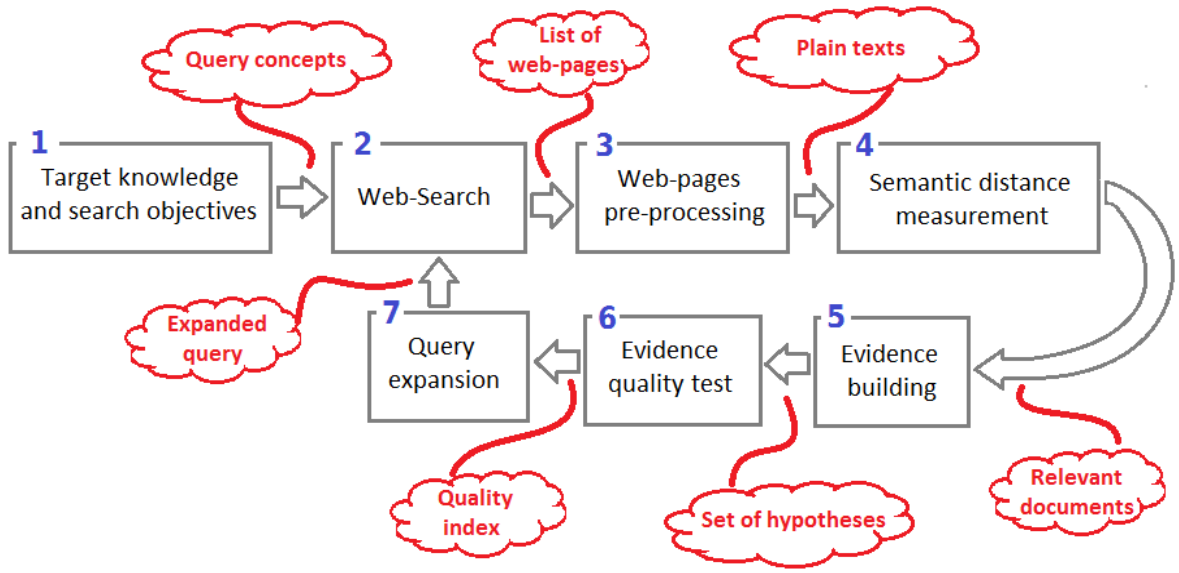


Figure 3.1 – Logic for the research developed algorithm

The overall search algorithm is iterative and can be split in two stages – preparation and search. The next two sections will explain the logic behind each of the algorithm steps in detail.

3.3 Preparation Stage

During the preparation stage, the search problem (or target information) has to be identified by analysis. Initial target knowledge and search objectives are identified manually by analysts and presented in an unstructured text format. This prepared text reflects the search topic and is used further in the algorithm as an initial set of KK (known knowns) and KU (known unknowns) – Rumsfeld's philosophy is being introduced here. A search query is compiled and a traditional search engine (such as Google) is then used to create a list of Web-search results.

3.3.1 Identify target knowledge and search objectives (Step 1)

Information search regardless of the environment follows a common set of actions. On any topic these actions involve a comparison of aspects of the information source with what is in the searcher's mind. The following diagram (Figure 3.2) schematically illustrates how the process of information acquisition is seen within this research.

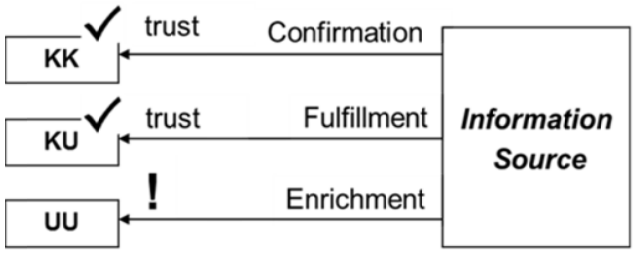


Figure 3.2 – Rumsfeld concepts within the process of information acquisition

No matter what the source of information is, online or printed, the individual performing the search always goes through the same information acquisition model. When analysing a new piece of information on any topic, he firstly pays attention on the things already familiar to him, in other words, his KK. The more of those he notices, the more reliable the source of information becomes to him. There will always be a comparison of the individual's current knowledge on the subject with what he is getting, and if there are any matches, the source of information will become more trustworthy to the individual.

It is likely that the analyst will discover information that is somewhat familiar, but knows very little about – KU. Through discovering more on the KU topic the previously unknown information converts to the "known" status and the individual's knowledge expands. As Chapter 2 explained, this means that the total knowledge remains the same and only the proportion of uncertainty on the topic has decreased.

It is also possible, that the analyst finds some information that was not known to exist. These pieces of information fall under the category of unknown unknowns (UU). This means of course, that discovery of UU changes the composition of the individual's uncertainty; ie, expanding the KU part of it. This can be seen as knowledge becoming wider and richer and previously unavailable option to look for the newly gained information is now available.

Therefore, every search process starts with understanding of the information need (KU) to formulate the search objective and identify which part of the discovered information is relevant. It is also important to understand the current available knowledge – this is closely related to the level of trust (or the level of the individual's confidence in a given piece of information). To summarise, the information discovery process comprises the following steps:

- Knowledge confirmation by hitting KK;
- Knowledge fulfilment by hitting KU;
- Knowledge enrichment by hitting UU.

The first two (confirmation and fulfilment) confirm the quality level of the information source and, therefore, increase the trustworthiness of this source. And the individual's initial knowledge on the search topic can be defined as the set of:

- KK – textual information that identifies the topic;
- KU – search objectives.

The algorithm proposed in this research will identify the available current knowledge on the topic as the collection of texts that comprise the key information setting the search topic (KK) along the free form text search objectives (KUs). This set of text (KK and KU) will be treated as the initial knowledge base for the proposed method.

Having set the textual representation of the current available, one can proceed to the next step of the algorithm – Web search.

3.3.2 Perform the Web-search (Step 2)

At this stage the algorithm, a search query that reflects the information need (KU) is formed manually. Figure 3.2 schematically demonstrates how a query and an initial knowledge base can be related to an individual's knowledge on a subject from the perspective of KKs and KUs.

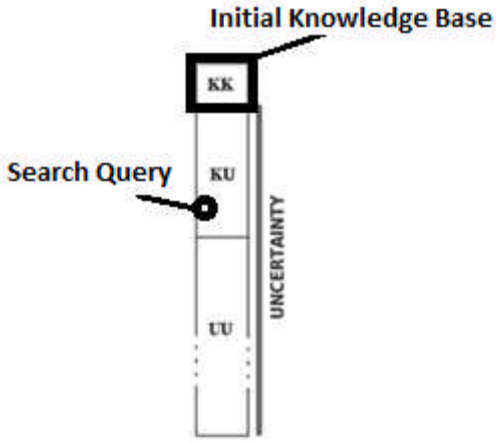


Figure 3.3 – An example of KK and KU within a query and an initial knowledge base

There are a number of available Web search engines that can process the query. Google search has been chosen for the purpose of this research as it employs the largest Web-page index. The aim of the proposed algorithm is to find as much semantically related information as possible in order to build not only accurate, but also complete evidence. Thus, one would rather remove unrelated content (false positives) in further steps than miss potentially relevant information (false negative).

The overall process is iterative and a single iteration corresponds to the agreed query run through the Google search. The resulting list of Web-pages then becomes an input for the next step of the algorithm – pre-processing.

3.3.3 Pre-process the Web-search results (Step 3)

Clearly, the Web contains a vast amount of valuable information. However, in practice, due to the complicated and flexible layout, the main content of a Web page is usually surrounded by various noisy information (such as advertisement, decoration, navigation bar/list, directory, menu, header, footer, etc.) as shown in figure 3.4. There is no doubt these elements play functionally important roles on the Web-page. But if not removed, they are big obstacles for automated information gathering and can seriously harm the accuracy of search. Therefore, extracting the main content of a Web page, especially main text, is a crucial pre-processing task.

Figure 3.4 – Example of noise objects on a Web-page

Hitherto, there have been a number of research projects conducted on eliminating noisy information from Web-pages (Zhou, Xiong, & Liu, 2009), (Adam, Bouras, & Poulopoulos, 2009), (Hu & Zhao, 2010), (Fu, Meng, Xia, & Yu, 2010), (Yi, Liu, & Li, 2009). This research employs Python library AlchemyAPI (www.alchemyapi.com) as it is a widely used powerful tool for accurate extraction of the main content from Web-pages ignoring menus, side bars, etc.

During the preparation stage, it is also important to remove stop-words (e.g. articles, prepositions, pronouns, etc.) that are frequent present in the natural language but hold very little semantic content – see Table 3.1.

Full text (stop-words highlighted)	Text with stop-words removed
Noise means **any** unwanted sound. Noise **is not** necessarily random. Sounds, particularly loud **ones, that** disturb people **or make it** difficult **to** hear wanted sounds**, are** noise. **For** example, conversations **of other** people **may be** called noise **by** people **not** involved **in any of them; any** unwanted sound **such as** domesticated dogs barking, neighbours playing loud music, portable mechanical saws, road traffic sounds, **or a** distant aircraft **in** quiet countryside, **is** called noise.	Noise means unwanted sound Noise necessarily random Sounds particularly loud disturb people difficult hear wanted sounds noise example conversations people called noise people involved unwanted sound domesticated dogs barking neighbours playing loud music portable mechanical saws road traffic sounds distant aircraft quiet countryside called noise

Table 3.1 – Example of stop word removal

There are several stop-word extraction techniques traditionally used in NLP applications. Below are the stop-word removal techniques which are considered classic and are widely used in various NLP tasks.

The most widely used stop-word list was proposed in (van Rijsbergen, 1979) where the author suggests a classic list of 250 stop-words in English. In (Luhn, 1958) the author defined a threshold below or above which a word could be labelled as a stop-word to determine the set of significant words in his document collection. In (Fox, 1990) the author formed a stop-word list of 421 words derived from the Brown corpus of 1,014,000 words drawn from a broad range of literature in English. SMART (SMART, 1974) is another example of a stop-word list used to remove common words.

Although, a stop-word list is usually domain dependent, the one formulated by Van Rijsbergen (which is also often used as a test baseline) is used for the purpose of this research.

Both – the initial knowledge base and the Web-pages returned by Google the search engine – need to be pre-processed with the above methods so that the textual content within both is clean and ready for the main part processing steps.

3.4 Search – Main Part

The main part (or search) may be considered as a combination of three integrated processes. Firstly, the application of NLP methods to enable the filtering of Web-search results to form a set of KK, KU and UU that are semantically related (close to the meaning of

the search topic), and thus, relevant to the search topic. This part of the algorithm combines statistical and semantic approach to analysing texts for semantic relatedness (or semantic closeness in this research). For each Web-page returned by a search engine, values for semantic closeness between its concepts and concepts in the initial knowledge base are calculated using distributional profiles of concepts as proposed in (Hirst & Mohammad, 2006).

Their method relies on a Roget style thesaurus (Roget, 2004) where words of English language are grouped into 1044 categories by the meanings of words. Having obtained the values for semantic closeness of concepts, the overall closeness score is calculated, using the technique by (Corley & Mihalcea, 2005), which was modified for this research. The original method used word-to-word similarity score for calculating closeness metric for two texts. In contrast, this research proposes using concept-to-concept similarity, in order to avoid ambiguity.

3.4.1 Perform semantic distance measurement (Step 4)

Traditional approach to finding the similarity between two text segments is to use a simple lexical matching method, and produce a similarity score based on the number of lexical units that occur in both input segments. In general, however, while successful to a certain degree, these lexical matching similarity methods fail to identify the semantic similarity of texts. A good part of the search problem lies in the diversity of the language that other people used to describe same things. For example,

"THE STOCK MARKET IS GOING DOWN" and
"THE SHARE MARKET IS CRASHING"

This example defines similar situations. Clearly, the traditional lexical matching will show poor results in identifying a score that indicates the similarity or closeness of the two pieces of text at the semantic level. So it is essential that an acceptable similarity test is identified.

Hitherto, there have been a number of knowledge-based measures that estimate semantic distance between words or concepts. These measures can be classified in two classes:

- Lexical resource based measures (Mihalcea, et al., 2006), those that rely on the structure of a knowledge source, such as WordNet;
- Distributional measures (Firth, 1957), those that rely on the distributional hypothesis, which states that two words are semantically close if they tend to occur in similar contexts.

The following two sections summarise the traditional lexical resource based measures and the distributional measures of semantic distance between words and concepts.

3.4.1.1 Lexical resource based measures

The most popular lexical source is WordNet (Fellbaum, 1998), used for various NLP tasks. WordNet is a lexical database of the English language. It consists of English Words, grouped into nodes (synsets) that represent concepts. Each synset is composed of a general definition and a set of synonymous words that share a particular concept. The synsets are connected by various semantic relations (such as hyponymy, meronymy, etc.).

An extensive survey of the various WordNet based measures is presented in (Mohammad & Hirst, 2006). The authors compared the measures, obtained with the use of WordNet, with human judgment on selected word pairs. The following section provides a brief summary of the major WordNet based measures of semantic distance.

1) Hirst-St-Onge

Nodes in a network may be connected by different kinds of lexical relations such as hyponymy, meronymy, and so on. The idea behind *Hirst-St-Onge* measure of semantic relatedness (Hirst & St-Onge, 1998) is that two concepts are semantically close, if their WordNet synsets are connected by a path that is not too long and that "does not change direction too often". If the starting point is a particular node c_1 and a path to c_2 lies via a particular relation (e.g. hyponymy), to a certain extent the concepts reached will be semantically related to c_1. However, if during the path, edges belonging to different relations are met (other than hyponymy), very soon words that are unrelated may be reached.

The strength of the semantic relationship is given by the formula:

$$rel(c_1, c_2) = C - PathLength - k * d, \tag{3.1}$$

where d is the number of changes of direction in the path, and C and k are constants; if no such path exists, $rel(c_1,c_2)$ is zero and the synsets are deemed unrelated.

2) Leacock-Chodorow

Leacock-Chodorow measure of similarity (Leacock & Chodorow, 1998) relies on the length of the shortest path between two WordNet synsets. In their approach only one relation type (hyponymy) is considered and the path length formula is modified to reflect the fact that edges lower down in the *is-a* hierarchy correspond to smaller semantic distance than the ones higher up.

The strength of the semantic relationship is given by the formula:

$$sim(c_1, c_2) = -\log \frac{len(c_1, c_2)}{2D}, \tag{3.2}$$

where $len(c_1,c_2)$ is the smallest number of WordNet links between the target concepts c_1 and c_2. The path length is scaled by the overall depth D of the taxonomy.

3) Resnik

Resnik approach (Resnik, 1995) suggests a combination of corpus statistics with WordNet. The idea behind the method is that the more general the lowest super-ordinate (*lso*) of the target nodes c_1 and c_2 is, the larger the semantic distance between the concepts, while the more general *lso* indicates the larger semantic distance between the nodes c_1 and c_2. This specificity is measured by the formula for information content (*IC*):

$$\begin{aligned} sim(c_1, c_2) &= IC(lso(c_1, c_2)) \\ sim(c_1, c_2) &= -\log p(lso(c_1, c_2)), \end{aligned} \tag{3.3}$$

where $lso(c_1,c_2)$ is the lowest super-ordinate of the target concepts c_1 and c_2 in WordNet.

Note, that the lower the *lso*, the lower the probability of occurrence of the *lso* and the concepts subsumed by it, and, hence, the higher its information content. In (Mohammad & Hirst, 2006) the authors mention that, as per Resnik's formula, given a particular lowest super-ordinate, the exact positions of the target nodes below it in the hierarchy do not have

any effect on the semantic similarity. Intuitively, one would expect that word pairs closer to the *lso* are more semantically similar than those that are distant.

4) Jiang–Conrath

Jiang–Conrath proposed a measurement of semantic distance, the inverse of similarity (Jiang & Conrath, 1997). The measurement uses the conditional probability of presence of an instance of a child synset given an instance of a parent synset. The information content of the two nodes plays a significant part, as well as that of their most specific subsumer. The similarity is measured with the formula:

$$sim(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(lso(c_1, c_2))}, \quad (3.4)$$

where $IC(c)$ is the information content denoted by $IC(c_i) = \log(p(c_1))$; and $lso(c_1,c_2)$ is the lowest super-ordinate of the target concepts c_1 and c_2 in WordNet.

3.4.1.2 Distributional measures

Distributional measures (Lee, 1999) provide metrics based on models of distributional similarity learned from large text collections (text corpora). Corpus based measures of distributional distance rely simply on text and can give the distance between any two words that occur at least a few times. Below is a brief summary of the major distributional measures of semantic distance. All of them follow the same two steps: (1) create distributional profiles, and (2) calculate the distance between two distributional profiles.

1) Cosine

The *Cosine* method (van Rijsbergen, 1979) is one of the earliest and most widely used distributional measures. This measure calculates the cosine of the angle between the distributional vectors of the two words w_1 and w_2. Figure (3.5) is a representation of a word w in the co-occurrence vector space. Values w_x, w_y, and w_z are its strengths of association with words x, y, and z, respectively. Figure (3.6) shows spatial distributional distance between target words w_1 and w_2.

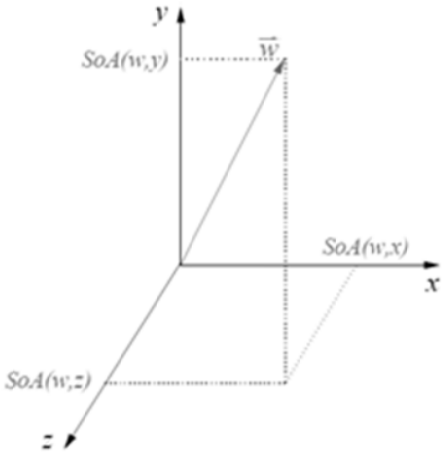


Figure 3.5 – A representation of a word ***w*** in the co-occurrence vector space

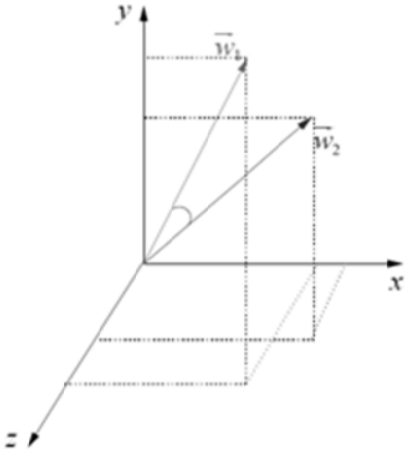


Figure 3.6 – Spatial distributional distance between target words w_1 and w_2

The more words co-occur with target words w_1 and w_2 (i.e the more context they share), the smaller the angle between their vectors becomes, therefore, increasing the cosine value between the two vectors, indicating high relatedness between the words:

$$Cos(w_1, w_2) = \frac{\sum_{w \in C(w_1) \cup C(w_2)} (P(w \mid w_1) \times P(w \mid w_2))}{\sqrt{\sum_{w \in C(w_1)} (P(w \mid w_1)^2} \times \sqrt{\sum_{w \in C(w_2)} (P(w \mid w_2)^2}}, \quad (3.5)$$

where $C(w)$ is the set of words that co-occur (within a certain text window) with the word w in a corpus. $P(w \mid w_i)$ is the conditional probability of the co-occurring words given the target word, and is used as the strength of association. The cosine measure gives scores in the range from 0 (distant/unrelated words) to 1 (synonymous).

2) Manhattan Distance and Euclidean Distance

Distance between any two words in the vector space can also be calculated using *Manhattan Distance* or *Euclidean Distance*. Manhattan distance (Lee, 1999) calculates the sum of the difference in strength of association of w_1 and w_2 with each word that they co-occur with (denoted by their conditional probability). The greater difference indicates the greater distributional distance between the two words and can be calculated using the formula:

$$MD(w_1, w_2) = \sum_{w \in C(w_1) \cup C(w_2)} |P(w|w_1) - P(w|w_2)|, \quad (3.6)$$

where $P(w|w_i)$ is the conditional probability of the co-occurring words given the target words, and is used as a measure of the strength of association.

Euclidean Distance measure (Lee, 1999) uses the root mean square of the difference in strength of association and is given by the formula:

$$ED(w_1, w_2) = \sqrt{\sum_{w \in C(w_1) \cup C(w_2)} (P(w|w_1) - P(w|w_2))^2}, \quad (3.7)$$

where $P(w|w_i)$ is the conditional probability of the co-occurring words given the target words, and is used as a measure of the strength of association.

Both Manhattan Distance and Euclidean Distance give scores in the range between 0 (synonymous) and infinity (distant/unrelated words).

3) Kullback-Leibler divergence

The relative entropy $D(p||q)$ of the given two probability mass functions $p(x)$ and $q(x)$ is:

$$D(p \| q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \quad \text{for } q(x) \neq 0 \quad (3.8)$$

Intuitively, if $p(x)$ is the accurate probability mass function corresponding to a random variable X, then $D(p||q)$ is the information lost when approximating $p(x)$ by $q(x)$. In other words, $D(p||q)$ is indicative of how different the two distributions are. Relative entropy is also called the *Kullback-Leibler divergence* (Kullback, 1997).

In (Mohammad, 2008) the author suggests using two probability mass functions $P(w|w_1)$ and $P(w|w_2)$, and calculates their relative entropy $D(p||q)$ as a measure of distributional distance, indicating how different the two distributions are:

$$KLD(w_1, w_2) = D(d_1 \| d_2) = \sum_{w \in V} P(w|w_1) \log \frac{P(w|w_1)}{P(w|w_2)} \text{ for } P(w|w_2) \neq 0$$

$$KLD(w_1, w_2) = \sum_{w \in C(w_1) \cup C(w_2)} P(w|w_1) \log \frac{P(w|w_1)}{P(w|w_2)}, \quad (3.9)$$

where V is the vocabulary (all the words found in a corpus) and $C(w_i)$ is the set of words co-occurring (within a certain window) with the word w_i. $P(w \mid w_i)$ is the conditional probability of the co-occurring words given the target words w_i, and is used as a measure of the strength of association.

It should be noted that the Kullback-Leibler distance is not symmetric, that is, the distance from w_1 to w_2 is not necessarily, and even not likely, the same as the distance from w_2 to w_1. The resulting distributional measure ranges from 0 (synonymous) to infinity (unrelated).

4) α-skew divergence

The *α-skew divergence* (Lee, 2001) is a slight modification of the Kullback-Leibler divergence:

$$ASD(w_1, w_2) = \sum_{w \in C(w_1) \cup C(w_2)} P(w \mid w_1) \log \frac{P(w \mid w_1)}{\alpha P(w \mid w_2) + (1-\alpha) P(w \mid w_1)}, \quad (3.10)$$

where α is a constant usually set to 0.99. $P(w \mid w_i)$ is the conditional probability of the co-occurring words given the target words w_i, and is used as a measure of the strength of association.

Analysis performed in (Lee, 2001) shows that α-skew divergence performs better than Kullback-Leibler divergence in estimating word co-occurrence probabilities. The resulting distributional measure ranges from 0 (synonymous) to infinity (unrelated).

5) Jensen-Shannon divergence

Jensen-Shannon divergence is another relative entropy-based measure (Dagan, Lee, & Pereira, 1994):

$$JSD(w_1, w_2) = D\left(d_1 \,\Big\|\, \frac{1}{2}(d_1 + d_2)\right) + D\left(d_2 \,\Big\|\, \frac{1}{2}(d_1 + d_2)\right) \quad (3.11)$$

$$JSD(w_1, w_2) = \sum_{w \in C(w_1) \cup C(w_2)} \left(P(w \mid w_1) \log \frac{P(w \mid w_1)}{\frac{1}{2}\left(P(w \mid w_1) + P(w \mid w_2)\right)} + \right.$$

$$\left(+ P(w \mid w_2) \log \frac{P(w \mid w_2)}{\frac{1}{2}\left(P(w \mid w_1) + P(w \mid w_2)\right)} \right) \tag{3.12}$$

Jensen-Shannon divergence is the sum of the Kullback-Leibler divergence between each of the individual co-occurrence distributions d_1 and d_2 of the target words w_1 and w_2 with the average distribution $((d_1+d_2)/2)$.

While the Kullback-Leibler distance is not symmetric, Jensen-Shannon divergence overcomes the symmetry problem. It also avoids the problem of zero denominators and, therefore, is always well defined.

The resulting distributional measure ranges from 0 (synonymous) to infinity (unrelated).

3.4.1.3 Issues with traditional measures

When used on their own, both lexical-resource-based and distributional measures of word-distance have significant limitations described in detail in (Mohammad & Hirst, 2006).

Among the lexical-resource-based methods, *Resnik* formula contradicts the intuitive fact that that word pairs closer to the lowest super-ordinate are more semantically similar than those that are distant. *Leacock-Chodorow* measure of similarity is limited only to the *is-a* type of relation between the WordNet nodes. Generally, for distance oriented measures computed on WordNet semantic network, as compared by (Budanitsky & Hirst, 2006), the Jiang-Conrath measure has the highest correlation (0.850) with the Miller and Charles list of 30 noun pairs (Miller & Charle, 1991) to which human judges had assigned similarity measures, and performs better than all other measures considered in a spelling correction task.

Overall, despite the fact that WordNet-based measures accurately estimate semantic similarity between nouns, their estimation of semantic relatedness especially in pairs other than noun–noun is extremely poor.

As for the distributional word-distance measures, the detailed survey in (Mohammad & Hirst, 2006) the authors concluded that Cosine method is the most accurate to estimate distributional distance between two concepts. It has the highest level of correlation with human rated word pairs of automatic rankings (Rubenstein & Goodenough, 1965).

However, distributional measures are expected to perform poorly in the face of word sense ambiguity if not enough context is present. Moreover, there is a data-sparseness problem associated with distributional measures, as they consider only the words that appear in the text, while ideally all the words in the vocabulary should be considered.

Finally, both WordNet-based and distributional measures require significant computational power to store and process extremely large matrices of size $N \times N$, where N is the size of the vocabulary.

Therefore, none of the above methods meet the requirement of the new solution to provide effective way of measuring semantic similarity between two texts.

3.4.1.4 Hybrid solution – distributional measurement of concept-distance

For the purpose of this research, a hybrid approach proposed in (Mohammad, 2008) was chosen. It combines the co-occurrence statistics of a distributional approach with the information in a lexical resource, and employs a distributional measure of concept-distance by calculating the distance between the distributional profiles of concepts rather than words. The distributional profile of a concept is the strength of association between it and each of the words that co-occur with it. The closer the distributional profiles of two concepts, the smaller is their semantic distance.

For the lexical resource (Mohammad, 2008) used published *Macquarie Thesaurus* concepts. This is a Roget-style thesaurus that, in contrast to any WordNet-style fine grained sense resources, classifies all English words into 1044 categories. The use of thesaurus categories as concepts allows pre-computing of all concept distance values required in a form of concept-concept distance matrix of size only 1044×1044 (about 0.01% the size of the matrix required by traditional lexical resource based and distributional measures). Another big advantage of the hybrid approach over the distributional measures is that the distance between two concepts (categories) is calculated from the occurrences of all the words listed under those categories. Considering all the advantages of the hybrid method by Mohammad (2008) it was chosen to best suit the task of this research. Next section will describe the algorithm of distributional measurement of concept-distance.

3.4.1.5 Algorithm of distributional measurement of concept-distance

The algorithm of the distributional measurement of distance between two concepts by (Mohammad, 2008) can be schematically represented as in the Figure 3.7 below.

Briefly, the algorithm can be described as follows. First, the text is parsed for the thesaurus categories and for each word co-occurring with the target word within a chosen text window its corresponding category is recoded. The aim of this step is to build a word-category co-occurrence matrix where the element m_{ij} corresponds to the number of times the word w_i was close to any words belonging to the category c_j within the chosen text window. This matrix allows performing bootstrapping and word-sense disambiguation.

Then a contingency table is created for each matrix element in order to calculate distributional profiles of concepts – vectors with coordinates corresponding to either point mutual information (PMI) (Ward Church & Hanks, 1990) or conditional probability values.

By applying the chosen distributional measure (see 3.3.1.2 – Distributional measures) the distributional distance between two target concepts can be calculated. Chapter 4 provides more details to the mathematics of the algorithm of measurement of concept distance using distributional profiles of concepts.

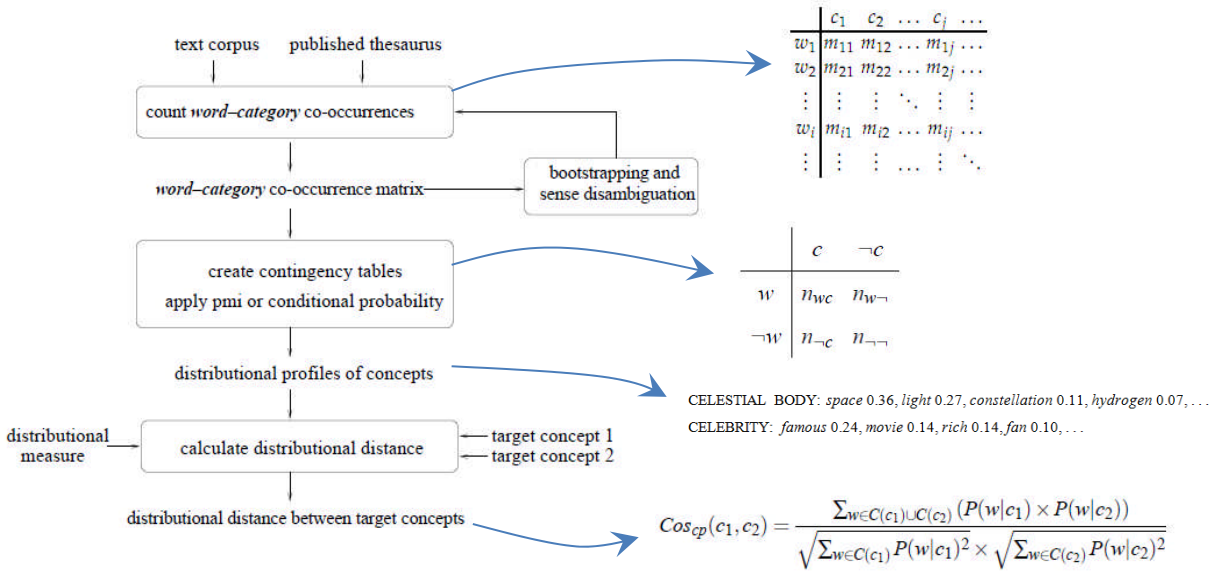


Figure 3.7 – Measurement of distance between distributional profiles of concepts

3.4.1.6 Measuring text similarity

Having obtained the results for semantic distances between concepts in two texts, the overall closeness score for the texts is calculated using the technique proposed by (Corley & Mihalcea, 2005) , which was modified for this research.

The original method measures semantic similarity between two texts by exploiting the information that can be drawn from the similarity of the component words, or word-to-word similarity. Given this metric and a measure of word specificity, this research defines the semantic similarity of two text segments T_1 and T_2 using a metric that combines the semantic similarities of each text segment in turn with respect to the other text segment.

The original method used word-to-word similarity; in contrast, this research modified this approach and relies on concept-to-concept similarity for calculating semantic closeness between two texts, in order to avoid ambiguity.

The resulting closeness score value lies between 0 and 1, with a score of 1 indicating identical text segments or duplicates, and a score of 0 indicating no semantic overlap between the two segments. Chapter 4 provides more details to the mathematics of the algorithm that measures semantic closeness between two texts.

Once the text similarity score is defined for all Web pages returned by the search engine, only those pages which have their extracted text similarity value above the chosen threshold are added to the existing knowledge base and go to the next step of the algorithm. By increasing the threshold only conceptually highly related Web-pages will be promoted to the next stage, while a lower threshold will result in more Web-pages further processed.

3.4.2 Build or Enhance Evidence (Step 5)

Evidential analysis is fundamental to the practice of intelligence analysis and requires the ability to represent, store, and manipulate evidence, hypotheses, and arguments. The chain of arguments, relating evidence to hypotheses, does not necessarily take from that hypotheses back to the initial evidence. The amount of information, that would be considered enough to build the evidence leading to sensible decision making, is unknown. At this stage

of the algorithm grounded theory is applied to test the completeness of gathered information up to and including the current iteration.

Grounded theory (Martin & Turner, 1986), (Corbin & Strauss, 2008) has been successfully used in building a hypothesis (theory) using interviews. Grounded theory is a systematic methodology in the social sciences involving the generation of theory from data. An important characteristic of grounded theory is that it does not use any prior information, and that it builds theory only based on information that is obtained throughout the research, making it very suitable for this research in the context of evidence building with very limited prior information. Other important properties are constant comparison of data with findings and theoretical sampling of different groups.

This part of research relies on the idea that only the information within the initial knowledge base is considered as known. Extracted text of every Web-page that has passed all previous filtering stages is compared against the knowledge base concepts and analysed for any new concepts. After the first iteration of the proposed algorithm, the newly collected evidence will partially consist of the text that is highly similar to the contents of the initial knowledge base. The major part of the new evidence will be new concepts. These new concepts are considered as converted to KU and KK. It is the rate of conversion into KK that this research is focused on. The comparison is done for the cumulative amount of new concepts calculated for previously checked Web-pages content added to the initial knowledge base.

Chapter 4 will explain the mathematics involved in the grounded theory when applied to the task of this research.

3.4.3 Test the evidence quality (Step 6)

Methods reviewed in Chapter 2 allow estimating relevance and completeness of search results. But again, only limited research has been carried out on metric for measuring the quality of the gathered information in order to know how good it is for decision making.

Whilst this research relies on Grounded theory to test the quantitative aspects of collected information, evidential analysis is applied to estimate its quality, thus, setting the quality parameter for the efficacy of the eventual decision-making. This research utilises Dempster-Shafer theory (Shafer, 1976) used to express uncertain judgements of experts, in this case to measure the quality level associated with gathered information.

The attraction for this theory over other formal techniques discussed in Chapter 2 is that it allows calculation of the quality score without any manual processing or pre-defining of data source models of any type, which makes Dempster-Shafer theory a powerful and coherent way of measuring information quality based on the semantic metric.

The qualitative relation between a piece of evidence and a hypothesis corresponds to a cause-consequence chain. A piece of evidence implies a hypothesis or a set of hypotheses, respectively. Dempster-Shafer theory requires that all hypotheses are unique, not overlapping and mutually exclusive. The strength of an evidence-hypothesis assignment is quantified by a statement of a data source, which in our case is a single Web-page.

A measure of strength of an evidence-hypothesis assignment is quantified by a statement of a data source (a single Web-page) and for this research it is propagated from the results of the semantic distance measure for each Web search result. After the semantic distance measurement stage, each Web-page's extracted text can be seen as a representation of thesaurus categories with Boolean values corresponding to whether the text includes any words from the category or not. In this research, the set of 1044 Thesaurus categories and all their possible combinations is a frame of discernment, or a power set. Thus, each text as a combination of Thesaurus categories is a member of the power set. Mass function for a text, as a proportion of all evidence that supports the text, is taken as a normalised value of the NLP analysis result – this will comply with the requirement of Dempster-Shafer theory for all mass functions to sum up to "1". The quality of information in each Web-page correlates with the Belief value for the corresponding text and reflects the amount of evidence one has for each Web-page.

Based on the obtained quality values for each Web-page text, the filtered search engine result list is reordered starting from those Web-pages with highest quality from semantic view point. These documents can be then transferred to a company's Intranet, where methods adapted to process finite set of documents would further analyse this new pool of information.

Details on mathematics for Dempster-Shafer approach to measuring quality are discussed in Chapter 4 of this thesis.

3.4.4 Identify further search direction (Step 7)

One of the issues associated with search expansion (see Figure 3.8) is keeping the search within the initial search topic boundaries in order to avoid returning information that is too remote. The discovered and converted UU are used as a basis for the new query expansion in order to identify the best information for the target decision process.

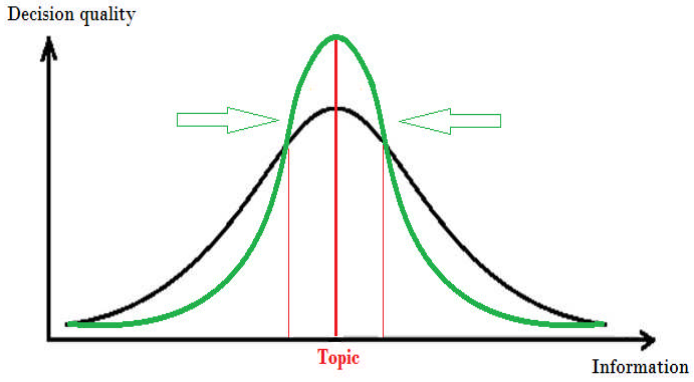


Figure 3.8 – Correlation between search topic boundaries and decision quality

Query expansion comes straight from the issue with topic boundaries. It is necessary to employ an effective method in order to improve the search results by expanding the search query to retrieve additional documents. For the purpose of this research, identification of additional topics and, hence, formulation of new search queries is done manually after revising the final results at the end of a single iteration.

3.5 Elements of the Proposed Algorithm

Below is the overview to the logic behind the algorithm. The overall process is semi-automatic and is further described in Chapter 4. The fundamental elements required for the algorithm to work are listed below.

- *Initial knowledge base* – all information currently available about the search topic in the form of unstructured text, i.e. natural language. The knowledge base is stored in a separate text file.

- *List of queries for search engine* – queries reflect the known unknowns about the search topic and define the direction of search.
- *Roget's thesaurus* – thesaurus of English words where all words associated with 1044 categories based on the meanings. Such categories represent different semantic concepts and each English word can be associated with more than one concept. The thesaurus is stored in a separate text file.
- *List of stop-words* – words that are used frequently in the language, e.g. articles, pronouns etc. List of stop-words is stored as a separate text file.
- *Search engine* – this research extensively uses Google search as holds the largest index of the Web. For this research it is not an issue to be overly inclusive during the first stages of the algorithm.
- *Python libraries and script files* – the proposed algorithm is evaluated with a semi-automated system realised with Python language.

The main module is the core module of the system and pre-processes the texts (knowledge base and Web-pages extracted text), calculates semantic distance between them, and calculates the KK conversion rate. Other modules are used to prepare the thesaurus text file and transform it to a collection of words/phrases with corresponding categories indexes, prepare text files (knowledge base and Web-pages extracted text) for further processing and analysis, calculate distribution profiles of concepts. For main content extraction from Web-pages ignoring side bars, menus etc. the AlchemyAPI library is used. In addition, NLTK library is extensively used as a powerful tool for processing raw text. Further information on the model software is provided in Chapter 5.

Firstly, all knowledge available on the search topic is collected, written in natural English language and recorded as a single text file. This forms up the initial knowledge base for the search. The list of associated queries for the search topic expansion is kept aside and each query is fed into the search engine separately. The above procedures are done manually and each query is associated with a single iteration of the overall search and analysis process.

Then, the knowledge base text is prepared for further analysis. Stop-words are removed in accordance with the pre-set stop-word list. Instances of words inside the text that relate to any of the 1,044 categories in the thesaurus form the Text-Concept Matrix (TCM). Based on the TCM, the Word-Concept Co-occurrence Matrix (WCCM) is built. Finally, using the contingency table to each unique word and category the Strength of Association (SOA) matrix is created, where each cell value is calculated as conditional probability $p(w|c)$

that a word *w* co-occurs with a concept *c* within the text window of ±5 words according to the contingency table.

Next, the search results for each query are processed and prepared to be fed into the system. For each Google results page, its source file is exported as a text file. The source code files are then merged into one so that the resulting file contains the html source code for all Google search results pages for the iteration. Next, a Python script is used to parse this text file and extract the html links to the Web-pages. Even though this technique of is manual and time consuming, it is the only possible universal way of scraping Google search results for the links to Web-pages due to the frequent changes in Google search system. The resulting text file contains a list of all html links to the Web-pages selected by Google in return to the query for the current iteration. The html links follow the same order as they would appear in the search engine results list.

After that, the resulting text file with the search results is fed into the system where the complex processing and analysis is done. Starting from the first line and moving down the list, each link to the Web-page is attempted to be opened, and if successfully done, the main content of the Web-page is extracted with the use of AlchemyAPI tool (www.alchemyapi.com). Then, the extracted text is pre-processed in the same way as the knowledge base (see above). Once the SOA matrix is created, everything is ready for the first part of the analysis – Natural Language Processing (NLP).

The NLP part calculates the semantic distance between two texts – the initial knowledge base and the extracted text for the Web-page currently processed. In the end, for each Web-page, the resulting semantic score is compared to the chosen threshold and only relevant texts (with semantic score above the threshold) will be passed through to the next stage of the algorithm. The NLP part starts with creation of an augmented list of unique words in both the knowledge base and the extracted text. This is done as a part of calculation of the Distributional Profiles (DP) of concepts – vectors that consist of SOA's between words and concept for a given text. The augmented list of unique words ensures the DP vectors for both texts are of the same dimension. Then, the similarity between each of the concepts in the two texts is calculated as Cosine of the angle between the two vectors. The results are recorded in a form of a matrix with dimension [1,044 x 1,044]. Having obtained concept-to-concept similarity values, the final semantic distance between the knowledge base and the Web-page can be calculated using the formula specially adapted for this research. The resulting value for the semantic score is between "0" and "1". The closer the semantic score value to "1", the closer the texts are in their meaning. If the extracted text of a Web-page text

is semantically close to the knowledge base by at least “0.85” (threshold), then this text is passed on to the next stage of the algorithm – the Grounded Theory analysis.

The Grounded Theory analyses the amount of new information coming in with the new text. In this research it is assumed that the topic is semantically exhausted if no new information can be added, of if new information does not significantly enhance the knowledge on the topic. In order to check this, the set of unique words in the initial knowledge base is considered as the set of KK, while the set of unique words within the extracted text of the Web-page is treated as KU and UU. The two sets are compared and those words that are not in the KK set are considered to be converted to KK and are added to the KK list. The next Web-page extracted text will be then compared against the updated list of KK. Once the number of the new words is recorded, the extracted text is passed onto the final stage of the algorithm – Evidential Analysis.

The Dempster-Shafer part calculates the quality of information that the Web-page contains. The approach to quality measurement is based on the semantic criteria and is realised with Dempster-Shafer method of judgement. The quality of information in each Web-page correlates with the Belief value for the corresponding extracted text and reflects the amount of evidence one has for each text directly. Based on the obtained quality values for each Web-page, the filtered search results list is reordered starting from the Web-pages of the higher quality from semantic view point.

The overall algorithm is presented in the Figure 3.9 below. The combination of the values for relevance, completeness, and quality of the resulting knowledge base after each iteration will help to determine the progress of the search.

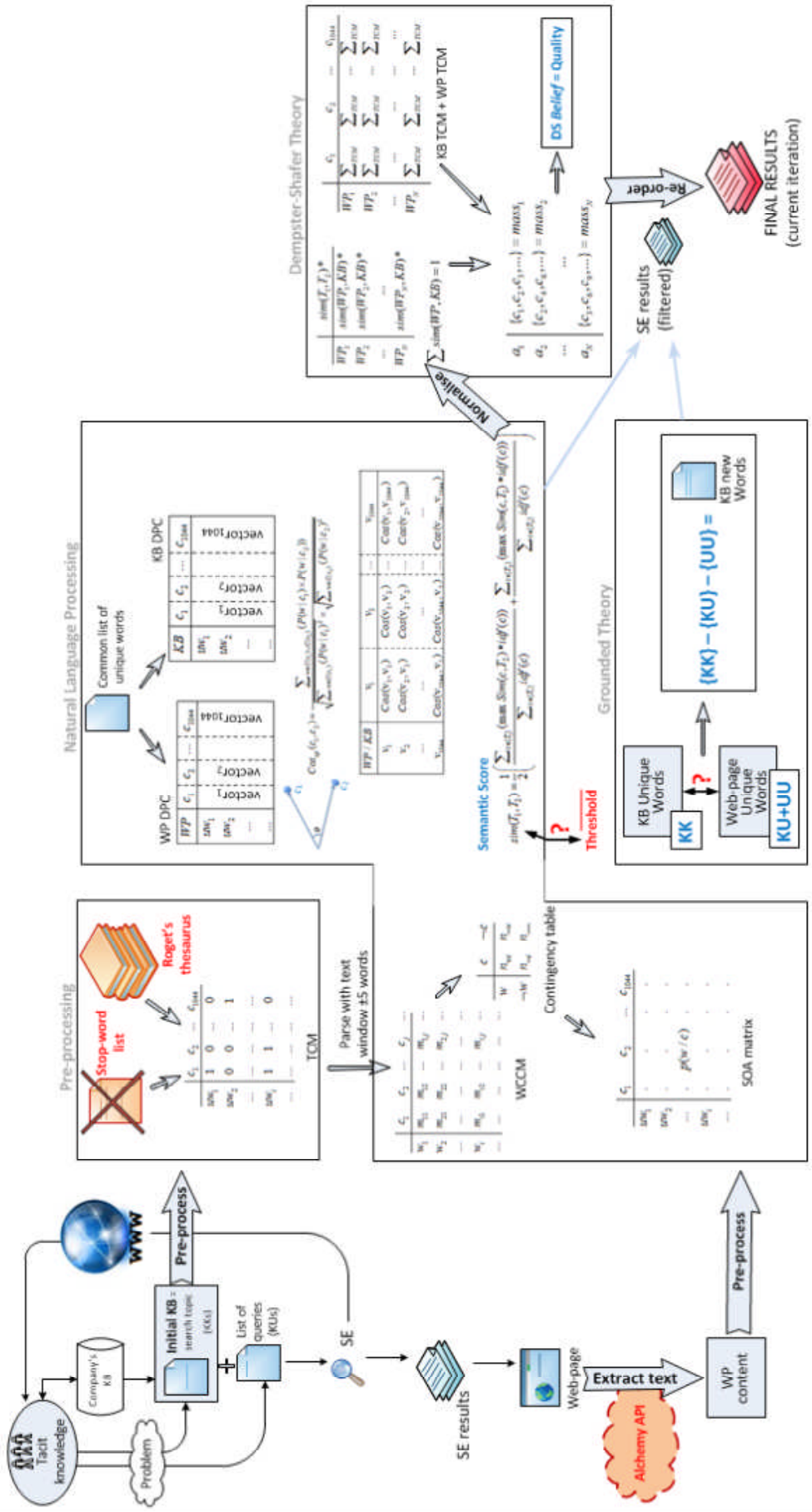


Figure 3.9 – Schematic Representation of the Approach of this Research

3.6 Summary

This thesis proposes a new approach that supports Web-based search of topic related information utilising the existing methods. In contrast to the currently available research methods, which tend to focus on specifically selected topics and information sources, this research presents a novel semantic approach method to Web-based discovery of previously unknown intelligence. The solution presented in chapter has no limitations for Web-sources and is open to perform search and analysis through any Web-page available to the search engine.

This chapter provided a comprehensive theoretical background to the proposed solution and discussed the details of its qualitative aspects. To summarise, the proposed algorithm suggests three stages of filtering. For each query, the natural language processing stage filters only those pages that are semantically relevant to the search topic. Then, the grounded theory part assesses the filtered Web-pages for holding new information and checks if this information exhausts the search topic. Lastly, the evidence on the Web-pages is analysed with Dempster-Shafer theory and its quality level is calculated as Belied function. Such filtering process permits to significantly reduce the length of the initial search results list returned by a traditional search engine, bringing the high-quality documents to the top of the list. The quality of collected information can be significantly enhanced by previously unknown information derived from the available known facts. This new information will expand the existing knowledge and form a knowledge base and can then be effectively manipulated by enterprise-search engine.

Chapter 4

Developing the Mathematics for the Search Algorithm

4.1 Overview to Chapter

The aim of this chapter is to address the research step:

Develop a search model that will integrate a conventional search engine with concepts and methods that will undertake the search

Following the overview description given and Chapter 3 of the algorithm being developed for the Web Search, this chapter provides detail on the mathematics in each stage of the algorithm. Particularly the chapter will further develop the ideas on how semantic closeness (section 4.3) of a Web-page's content is assessed against the initial search topic – this idea is at the heart of the algorithm. Furthermore, the chapter will develop mathematics for application of grounded theory (section 4.4) to measure completeness of the collected information. Moreover, it explains how this information is tested for quality using Dempster-Shafer theory (section 4.5). However, to ensure that the algorithm is set in the correct context it is important that some points from Chapter 2 are revisited as preparation (section 4.2). It is also necessary to focus on how a decision is made for iteration (section 4.6). By the end of this chapter a comprehensive description of all mathematics used for the foundation of the search model (algorithm) and hence the description of the algorithm will be complete and ready for a computing software representation to de developed (Chapter 5). Proof of used mathematics is attached in Appendix 6.

4.2 Preparation Discussion

Recalling from Chapter 2, the traditional approach to Web-search is based on indexing of the Web content, i.e. building an index database, and then searching for the keywords that match the content of this database. However, this strategy will not easily

support intelligence information acquisition. The Google search engine (the most commonly used) is able to find several millions of Web-pages and display up to 1000 results for a particular search in a fraction of a second, but these pages are not necessarily semantically related to the search topic. Even though Google currently has the best duplicate content filtering technology (Pugh & Henzinger, 2001) and (Gomes & Smith, 2000), it cannot analyse the meaning of the texts to eliminate semantically repeated documents, quantity does not always mean quality.

The keyword matching search technique essentially misses important information, while ranking strategy may place irrelevant search results at the top of the list. What should also be borne in mind is that the keyword being used reflects what the author has in mind and not necessarily what is required by the intelligence search, resulting in possible relevant information being missed.

A recently suggested approach to overcome this information problem is “concept search”, i.e. analysis of unstructured (plain) text for information that is conceptually similar to the information provided in a search query; ideas expressed in the retrieved information are relevant to the ideas in the text of the search query. Concept search is widely used in enterprise-search and data management systems, such as Autonomy (Autonomy, 2009), that operate with the finite knowledge base, making it possible to “understand” the meaning of the short query by extracting the meaning of the documents that are currently opened on an analyst’s PC desktop. Regardless of the effectiveness of such methods in the Intranet environment, Web scale far exceeds the amount of information that these methods can process reasonably in a realistic time frame.

This research model employs an existing search engine – Google. The aim is to build not only accurate, but also complete evidence; the search engine should not skip a Web source because it is not in its index base. It is more prudent to filter unrelated text at a later stage. For each query (manually created) the search results list is semi-automatically processed and prepared to be fed into the system.

The initial knowledge base and search objectives are identified manually and presented in unstructured text format. Clearly, the Web contains a vast amount of valuable information. However, in practice, due to the complicated and flexible layout, the main content of a Web-page is usually surrounded by noisy information (such as menu, header, advertisement, etc.). Therefore, extracting the main text of a Web-page is a critical processing task, if relevant information is to be identified.

The initial knowledge base and the collection of extracted texts from Web-search results are pre-processed as part of semantic analysis. Each Web-page from the list is accessed and run through AlchemyAPI tool in order to eliminate noise and extract main textual content. Once the text has been extracted, stop-words are removed (in accordance with the selected stop-word list) leaving a "clean" collection of Web-page's terms as an input for the next pre-processing step. For the next stage of the algorithm – semantic closeness measurement – this research uses a hybrid method detailed in (Hirst & Mohammad, 2006) that combines the co-occurrence statistics with the information in a lexical source. For the lexical source this research uses Roget's Thesaurus (www.roget.org) that, in contrast to traditionally used WordNet (Fellbaum, 1998), classifies all English words into 1044 categories. Words within the text correspond to concepts, or meaning categories, in the thesaurus. The Text-Concept Matrix (TCM) is then built. Rows of this matrix are associated with the unique words within the text, i.e. duplicates or the word order is not considered in this case. The columns represent 1044 concepts of the thesaurus – see Figure 4.1.

	c_1	c_2	...	c_{1044}
uw_1	m_{11}	m_{12}	...	m_{1j}
uw_2	m_{21}	m_{22}	...	m_{2j}
...	...	...	...	...
uw_n	m_{i1}	m_{i2}	...	m_{n1044}

Figure 4.1 – Example of the Text-Concept Matrix

Thus, the TCM is of dimension [n_u x 1044], where n_u is the amount of unique words within the text. TCM values m_{ij} are either "1" indicating that the word is associated with the concept in the thesaurus, or "0" in the other case.

4.3 Semantic Closeness Measurement

Based on the TCM (Figure 4.1), a Word-Category Co-occurrence Matrix (WCCM) is built and populated with co-occurrence counts from the text (Figure 4.2). Rows of this matrix are associated with the words as they appear in the text. In order to build the WCCM, the text should be parsed for the words that are positioned within the text window of ±5

words to the target word. It is created having word types '*w*' as one dimension and thesaurus categories '*c*' in the other. The WCCM is of the type:

	c_1	c_2	...	c_j	...
w_1	m_{11}	m_{12}	...	m_{1j}	...
w_2	m_{21}	m_{22}	...	m_{2j}	...
...	...	...	...	...	...
w_i	m_{i1}	m_{i2}	...	m_{ij}	...
...	...	...	...	...	...

Figure 4.2 – Word-Category Co-occurrence Matrix (WCCM)

A matrix cell m_{ij}, corresponding to word w_i and concept c_j, is populated with the number of times w_i co-occurs with any word listed under concept c_j in the thesaurus within a text window of ±5 words. Thus, for each word in the text 5 words to the left and 5 words to the right from this target word are remembered.

The values for these neighbouring 10 words are then found in the TCM and their TCM rows are summed up and added as the row which corresponds to the target word. The resulting matrix is of the size [*N* x 1044], where *N* is the number of all words in the texts excluding stop-words. Each value c_{ij} corresponds to a number of times a word *i* co-occurred with any word listed under the category *j* within the text window of ±5 words.

A contingency table (Figure 4.3) is then generated for every word *w* and category *c* via merging the cells for all other words and categories and adding up their frequencies.

	c	$\neg c$
w	n_{wc}	$n_{\neg w}$
$\neg w$	$n_{\neg c}$	$n_{\neg\neg}$

Figure 4.3 – Contingency Table

Finally, a conditional probability will yield the values for the strength of association between the target concept and co-occurring words in form of distributional profile of a concept:

$$P(w \mid c) = \frac{P(w \cap c)}{P(c)} = \frac{n_{wc}}{n_{\neg c}}. \qquad (4.1)$$

For example, consider the word STAR appearing in two texts. In text_1 it co-occurs within a text window with words belonging to thesaurus categories "*space*", "*light*", "*constellation*" and "*hydrogen*". In text_2 it is positioned next to words from categories "*famous*", "*movie*", "*rich*" and "*fan*".

Below are example distributional profiles of two senses of the word STAR (Mohammad, 2008) :

CELESTIAL BODY: *space* 0.36, *light* 0.27, *constellation* 0.11, *hydrogen* 0.07, ...
CELEBRITY: *famous* 0.24, *movie* 0.14, *rich* 0.14, *fan* 0.10, ...

In this example, the distributional profiles of concepts technically represent vectors of the size 1044, i.e. containing values that correspond to all 1044 thesaurus categories, as shown in Table 4.1.

Text Concepts	**Thesaurus Categories**									
	…	*constellation*	*famous*	*fan*	*hydrogen*	*light*	*movie*	*rich*	*space*	…
CELESTIAL BODY	…	0.11	0	0	0.07	0.27	0	0	0.36	…
CELEBRITY	…	0	0.24	0.10	0	0	0.14	0.14	0	…

Table 4.1 - Example distributional profiles

The Strength of Association matrix is then created (figure 4.4), where each entry is calculated as conditional probability $p(w|c)$ that a word w co-occurs with a concept c within the text window of ±5 words as from the contingency table.

	c_1	c_2	...	c_{1044}
w_1	$p(w_1 \mid c_1)$	$p(w \mid c)$	...	$p(w_1 \mid c_{1044})$
w_2	$p(w_2 \mid c_1)$	...	...	...
...	...	...	...	...
w_N	$p(w_N \mid c_1)$	....	...	$p(w_N \mid c_{1044})$

Figure 4.4 – Strength of Association Matrix

Based on the detailed survey of semantic distance measures in (Hirst & Mohammad, 2006), this research uses the adapted Cosine method to estimate distributional distance between two concepts. The choice of the Cosine concept distance measurement (with conditional probability as the strength of association in distributional profiles of concepts) was made based on the highest level of correlation with human rated word pairs of automatic rankings (Rubenstein & Goodenough, 1965). The Cosine distributional distance measure is denoted by:

$$Cos(c_1, c_2) = \frac{\sum_{w \in C(c_1) \cup C(c_2)} (P(w \mid c_1) \times P(w \mid c_2))}{\sqrt{\sum_{w \in C(c_1)} (P(w \mid c_1)^2} \times \sqrt{\sum_{w \in C(c_2)} (P(w \mid c_2)^2}}, \quad (4.2)$$

where $w \in C(c_1) \cup C(c_2)$ is the set of words that co-occur with concepts c_1 and c_2 within a text window of ±5 words in both texts. Thus, the formula for $Cos(c_1, c_2)$ measures semantic distance between each concept in each text, and treats the distributional profiles of concepts as vectors of the size equal to the number of all unique words in both texts. $P(w \mid c_1)$ and $P(w \mid c_2)$ are conditional probabilities of a word *w* co-occurring with any word listed under the category *c* in the thesaurus. Conditional probabilities are used as strengths of association between each word and each concept in both texts, and are taken from the distributional profiles of concepts. The value for the cosine in this case lies between 0 and 1, indicating semantic remoteness of two concepts when the value approaches 0 and semantic closeness when the value is close to 1.

The use of thesaurus categories as concepts allows pre-computing of all concept distance values required in a form of concept-concept distance matrix of a size much smaller than word-word distance matrix.

Having obtained the concept distances, the algorithm then calculates semantic closeness score between the initial knowledge base text and the extracted text of the Web-

page. The formula for measuring similarity between texts, proposed by (Corley & Mihalcea, 2005) was adapted for this research.

The original method measured semantic similarity of texts by exploiting the information that can be drawn from the similarity of the component words. First, for each word *w* in the text T_1 the word in the text T_2 that has the highest semantic similarity (max*Sim*(*w*; T_2)) is identified, according to the word-to-word similarity measure. Next, the same process is applied to determine the most similar word in text T_1 in relation to the words in texts T_2. The word similarities are then weighted with the corresponding word specificity, summed up, and the resulting similarity scores are combined using a simple average.

In contrast to the original approach, this research adapts the method by introducing concept-to-concept instead of word-to-word distance for measuring semantic similarity between two texts.

By applying the formula (4.2) the Cosine distributional distance values $Cos(c_1, c_2)$ are calculated between all concepts in two texts, forming the Cosine matrix of the size [1044x1044], the values represent semantic distances between the concepts (figure 4.5):

	$c_1(T_2)$	$c_2(T_2)$	...	$c_{1044}(T_2)$
$c_1(T_1)$	$Cos(c_1(T_1), c_1(T_2))$	$Cos(c_1(T_1), c_2(T_2))$	...	$Cos(c_1(T_1), c_{1044}(T_2))$
$c_2(T_1)$	$Cos(c_2(T_1), c_1(T_2))$	...	...	...
...	...	...	...	...
$c_{1044}(T_1)$	$Cos(c_{1044}(T_1), c_1(T_2))$	....	...	$Cos(c_{1044}(T_1), c_{1044}(T_2))$

Figure 4.5 – Cosine matrix for two texts

Given a measure for semantic distance between each of the concepts in both texts, it is possible to define the semantic similarity of two texts.

First, for each concept $c_i(T_1)$ in the text T_1 the concept $c_j(T_2)$ in the text T_2 is identified that has the highest semantic similarity $Cos(c_i(T_1), c_j(T_2))$, which will be denoted as $\max Sim(c, T_2)$. The concept similarities are then weighted with the corresponding concept inverted document frequency $idf(c)$, that has the value of 1 if the concept *c* is used in both texts, and 0.5, if the concept is used only in one of the two texts. This allows calculating a metric that combines semantic similarity of concepts in text T_1 with respect to concepts in text T_2:

$$\frac{\sum_{c\in\{T_1\}}(\max Sim(c,T_2)*idf(c))}{\sum_{c\in\{T_1\}}idf(c)}. \tag{4.3}$$

Applying semantic similarity of concepts in text T_2 with respect to concepts in T_1:

$$\frac{\sum_{c\in\{T_2\}}(\max Sim(c,T_1)*idf(c))}{\sum_{c\in\{T_2\}}idf(c)}. \tag{4.4}$$

The similarity between two texts T_1 and T_2 is therefore determined using the following function:

$$sim(T_1,T_2)=\frac{1}{2}\left(\frac{\sum_{c\in\{T_1\}}(\max Sim(c,T_2)*idf(c))}{\sum_{c\in\{T_1\}}idf(c)}+\frac{\sum_{c\in\{T_2\}}(\max Sim(c,T_1)*idf(c))}{\sum_{c\in\{T_2\}}idf(c)}\right). \tag{4.5}$$

To summarise, for each concept *c* in the initial knowledge base text T_1 the concept in the Web-page text T_2 is identified that has the highest semantic similarity $\max Sim(c,T_2)$, according to the concept-to-concept similarity $Cos(c_1,c_2)$ described above. Next, the same process is applied to determine $\max Sim(c,T_1)$ for the most semantically close concepts in T_2 compared to the concepts in T_1. The concept similarities are then weighted with the corresponding concept inverted document frequency $idf(c)$. Finally, concept similarities are summed up, and the resulting scores are combined using an average.

The semantic closeness $sim(T_1,T_2)$ has value between 0 and 1; the closer the value is to 1, the closer the texts are in their meaning. If the Web-page text is semantically close to the initial knowledge base text by at least 85% threshold, then the Web-page text is passed on to the next stage of the algorithm – the Grounded theory analysis.

4.4 Completeness Measurement

At this stage the Web-pages that were selected during the previous stage are tested for completeness – value they would add if they were to be added to the initial knowledge base. The information search process is independent of the search environment and comprises the same actions. On any topic these actions involve a comparison of content of the information source with information that is already known (KK and KU) and discovery of

UU, i.e. a comparison of currently known knowledge with new information retrieved (see Chapter 2 for details). The more matches observed, the more reliable and trustworthy the source of information becomes. The combination of KU and UU represents the uncertainty on the topic. Information discovery changes the level of uncertainty and its composition in an individual's knowledge by converting unknown information into known information. Following the first iteration of the algorithm, newly collected evidence will partially consist of the text that is similar to the contents of the initial knowledge base, while the major part of the new evidence will be new concepts.

The volume of information that would be considered enough to build the evidence leading to sensible decision making is unknown. However, for the purpose of this research it is assumed that the topic is exhausted if no new information can be added, or if new information does not significantly enhance the knowledge known on the topic. Grounded theory (Martin & Turner, 1986), (Corbin & Strauss, 2008) has been successfully used for building a hypothesis (theory) using interviews. Grounded theory is a systematic methodology from the social sciences involving the generation of intelligence evidence from data. An important characteristic of grounded theory is that it does not use any prior information, and that it builds theory only based on information that is obtained throughout the research, making it suitable in the context of evidence building with very limited prior information.

This stage of the algorithm employs grounded theory to analyse the amount of new information coming from the Web-pages and to test the completeness of gathered information. Only the information from the initial knowledge base is classified as known. It is also assumed that the search topic is complete when new search does not add any new information.

Every Web-page that is considered relevant at the previous stage of the algorithm is compared to the knowledge base and analysed for the amount of new concepts within the text of this Web-page. Grounded theory is an integral part in the proposed approach to identify the set of UU in newly gathered information through comparison of the conversion rate of KU and UU (new concepts) into KK (evidence). Total knowledge on a topic K_{total} is the collection of all three sets. It is the sum of initial knowledge base concepts KK_0, initial search objective concepts KU_0, while 'unknown unknowns' UU_0 are undefined:

$$K_{total}(0) = KK_0 + KU_0 + UU_0. \tag{4.6}$$

Following each iteration, newly identified concepts are added to the initial knowledge base, thus expanding the evidence:

$$\mathrm{K}_{total}(i) = \mathrm{K}_{total}(i-1) + \mathrm{KU}_i, \tag{4.7}$$

where KU represent new concepts on each iteration, and KU(i–1) ≠ KU(i). A change in KU represents the conversion rate $\delta(KU)$ of new concepts in evidence and is defined as:

$$\delta(KU) = \mathrm{KU}_i - \mathrm{KU}_{(i-1)}. \tag{4.8}$$

If $\delta(KU) > 0$, then there are still possible concepts that can be identified for evidence expansion. If $\delta(KU) < 0$, then no new information is being identified and one can assume that the topic is tending to exhaustion. Conversion rate $\delta(KU)$ is used to analyse the change in new concepts after each iteration. The comparison is done for the cumulative amount of new concepts calculated for previously checked Web-pages' content added to the initial knowledge base. In conjunction with quality test (next stage), conversion rate provides a basis for the decision of acceptance of the Web-page content. When $|\ \delta(KU)\ | \approx 0$ and the conversion function converges to a number, it indicates that the search topic is exhausted. Having identified the effect of the information from the Web-page on the completeness of the initial knowledge base, quality of the Web-page is then tested.

4.5 Quality Measurement

Evidential analysis is fundamental to the practice of intelligence analysis and requires the ability to represent, store, and manipulate evidence. This research applies evidential analysis to estimate the quality of collected information, hence, setting a quality parameter for the efficacy of the eventual decision-making. Dempster-Shafer theory of evidence (Dempster, 1968), (Shafer, 1976) has been chosen for a well-understood, formal framework for judging the evidence under uncertainty. The mathematical connection between information retrieval and Dempster-Shafer theory was suggested by Van Rijsbergen (1992). Dempster-Shafer theory (DS) is utilised for measurement of the quality level associated with gathered information. The quality is measured considering semantic closeness values only. However, there is an option to expand the scope of the quality metric and broaden its parameters by including some Web-site statistics, as proposed in (Danilova & Stupples,

2012). The total quality score for each Web-page is calculated as a Belief function; the higher the value of Belief for a Web-page, the higher its quality value and the higher the Web-page will be ranked in the final list of results. Next, the main concepts of DS theory are described as presented by Shafer (1976) and related with the context of Web-based discovery of unknowns.

4.5.1 Frame of discernment

In the Dempster-Shafer theory propositions are represented as subsets of a given set. The hypotheses (in context – texts as combination of concepts) represent all the possible states of the system considered. It is required that all hypotheses are elements (singletons) of the frame of discernment, which is given by the finite universal set *U*. The set of all subsets of *U* is its power set 2^U. A subset of those 2^U sets may consist of a single hypothesis or a conjunction of hypotheses. Moreover, it is required that all hypotheses are unique, not over-lapping and mutually exclusive.

If the value of some quantity is *u*, and the set of its possible values is *U*, then the set *U* is called a frame of discernment. "The value of *u* is in A" is a proposition for some $A \subseteq U$. The proposition $A = \{a\}$ for $a \in U$ constitutes a basic proposition "the value of *u* is *a*". Thus, each text can be represented as a set of Thesaurus categories, where each element is a Boolean value corresponding to whether the text includes any words from the category. Hence, the set of 1044 Thesaurus categories and all their possible combinations is a universal set $\underline{U} = \{c1, \ldots, c1044\}$. Each Web-page text as a combination of Thesaurus categories is a member of the power set $2^U = \{\emptyset, \{text\_1\}, \{text\_2\}, \{text\_3\}, \ldots, U\}$.

4.5.2 Basic probability assignment

In order to express the uncertainty of propositions, Beliefs can be assigned to them. The Beliefs are usually computed using a basic probability assignment (*bpa)* or mass function $m: 2^U \rightarrow [0,1]$. It has two properties: the mass of the empty set is zero, and the masses of the remaining members of the power set add up to a total of 1:

$$m(\emptyset) = 0 \text{ and } \sum_{A \subseteq U} m(A) = 1. \quad (4.9)$$

Mass function $m(A)$ expresses the proportion of all relevant and available evidence that supports the claim that the actual state belongs to A but to no particular subset of A. The value of $m(A)$ pertains only to the set A and makes no additional claims about any subsets of A, each of which have, by definition, their own mass. If there is positive evidence for the value of u being in A then $m(A) > 0$, and A is called a focal element. No Belief can ever be assigned to a false proposition. The focal elements and the associated *bpa* define a body of evidence. The mass function for a text, as a proportion of all evidence that supports the text, is a normalised value of the NLP analysis result according to $sim(T_1, T_2)$. The calculated semantic distance has a value between 0 and 1, and the total evidence is scaled to fall between 0 and 1 in order to satisfy the definition of *bpa*.

4.5.3 Belief function

Given a body of evidence with m, the total Belief can be computed provided by that body of evidence for a proposition. This is done with a Belief function *Bel:* $2^U \rightarrow [0, 1]$ defined upon m. Probability values are assigned to sets of possibilities and Belief in a hypothesis is constituted by the sum of the masses of all sets enclosed by it:

$$Bel(A) = \sum_{B \subseteq A} m(B). \tag{4.10}$$

$Bel(A)$ is the total Belief committed to A, i.e. the $m(A)$ itself plus the mass attached to all subsets of A. $Bel(A)$ is then the total positive effect the body of evidence has on the value of u being in A. The quality of information in each text is associated with the value of the Belief function for corresponding texts and reflects the amount of evidence supporting each text directly. Each text can be seen as a combination of Thesaurus concepts. For example, the Belief in the text_1 as a set of concepts {c1 U c2} is the sum of its own basic assignment with those of all of its subsets. Thus, if text_1 = {c1, c2}, text_2 = {c1} and text_3 = {c2}, the Belief function of the text_1 is:

$$Bel(text\_1) = m(text\_3) + m(text\_2) + m(text\_1). \tag{4.11}$$

If Table 4.2 below showed the distribution of concepts and the corresponding semantic similarity scores $sim(T_1, T_2)$, the example mass functions would represent the normalised values of $sim(T_1, T_2)$:

Text	**Thesaurus Categories**					$sim(T_1, T_2)$	$m(text)$
	C_1	***C_2***	***C_3***	**…**	***C_1044***		
Text_1	x	x			x	0.30	0.11
Text_2	x					0.50	0.19
Text_3		x	x			0.85	0.31
Text_4			x		x	0.40	0.15
Text_5	x				x	0.65	0.24

Table 4.2 – Example set of texts as input for calculation of Belief function

Therefore, from the formula (4.11) the Belief value for Text_1 $Bel(Text\_1)$ can be calculated as:

$$Bel(Text\_1) = m(Text\_1) + m(Text\_2) + m(Text\_5)$$
$$Bel(Text\_1) = 0.11 + 0.19 + 0.24 = 0.54$$

The value “0.54” represents the quality of the *Text*_1 form semantic viewpoint.

Based on the results of tests for completeness and quality, a decision is made on whether to iterate or stop.

4.6 Decision on the Next Iteration

The Web-pages with the highest quality from semantic view point are placed at the top of the list. The Web-pages with a quality value of 10% and higher that have passed through the grounded theory test will be suggested for transferring to the organisation's knowledge base. And those Web-pages with quality value over 85% will be considered as

"high quality" information. At this stage, the list of Web-pages suggested by the search engine is significantly reduced. It contains only Web-pages that have passed all three main filtering stages of the algorithm. The pages are highly relevant to the subject of the initial knowledge base. They contain new information that will expand the knowledge on the search topic, and will be ordered based on the quality of collected evidence rather than popularity. The decision on whether to iterate or stop is based on evidence tests for completeness and quality. Table 4.3 shows all possible combinations of results from the grounded theory test and the evidence quality test.

Conversion rate, $\delta(KU)$	**Quality**	**Action**
Positive	Positive	Continue (expand query)
Positive	Negative	Stop searching
Negative	Positive	Continue (expand query)
Negative	Negative	Stop (change formulation)

Table 4.3 – Possible combinations of outcome and decision on the next iteration

Thus, the decision on the next iteration depends on the amount of the discovered and converted UU coming into the knowledge base as well as the change in quality of knowledge base, if new information is to be added. The overall process is iterative and applied to the Web. The discovered unknowns are used as a basis for new query expansion in order to identify the best information for the target decision process.

4.7 Summary

The new approach for the search algorithm proposed in this thesis, employs the combination of existent Natural Language Processing techniques, grounded theory and Dempster-Shafer theory to automatically extract unknown unknowns from Web-based textual content and enrich the knowledge base with relevant and high quality information. This chapter addressed quantitative details of each stage of the proposed algorithm and provided mathematical background of the methods employed.

Regarding the Natural Language Processing part of the proposed search algorithm this chapter has developed mathematics for the following parameters:

- a Text-Concept matrix (TCM);
- a Word-Category Co-occurrence Matrix (WCCM);
- a distributional profile of a concept;
- a Strength of Association matrix (SOA);
- a Cosine concept distance measurement ($Cos(c_1, c_2)$);
- a semantic similarity score between two texts ($sim(T_1, T_2)$).

The following mathematics was developed on the application of Grounded Theory for completeness assessment of the gathered information:

- the total knowledge on a topic (K_{total});
- the conversion rate of unknowns ($\delta(KU)$).

As part of evidential analysis, Dempster-Shafer theory was adapted for the quality measurement of the gathered information from semantic view point. Mathematics for the following parameters was developed for this stage:

- a power set (2^U);
- a basic probability assignment or a mass function ($m(A)$);
- a Belief function ($Bel(A)$).

Therefore, mathematics behind the semantic analysis explained how to select Web-pages that are relevant to the initial knowledge base content. This research uses a hybrid approach that combines the co-occurrence statistics with the information in Roget's Thesaurus. The distributional measure of concept-distance is calculated as distance between the distributional profiles of concepts – strength of association between a target concept and each of the words in its context. Having obtained the concept distances between the initial knowledge base text and the extracted text of the Web-page, the algorithm then calculates semantic closeness score for the two texts.

Every Web-page that is considered relevant at the previous stage of the algorithm is compared to the knowledge base and analysed for the amount of new concepts within the text

of this Web-page. Grounded theory is used to test the completeness of the knowledge base – for each iteration a conversion rate is used to analyse the change in new concepts. Once the effect of the information from the Web-page on the completeness of the initial knowledge base is identified, quality of the Web-page is then tested.

Dempster-Shafer theory is used to estimate the quality of Web-pages content which is correlated with value of Belief function. The Web-pages with a quality value of 10% and higher that have passed through the grounded theory test will be suggested for transferring to the organisation's knowledge base. And those Web-pages with quality value over 85% will be considered as “high quality” information.

The combination of results from completeness and quality tests is compared against the table of possible outcomes for these two stages. The decision on the next iteration depends on the amount of new concepts coming into the knowledge base as well as the change in its quality, if new information is to be added.

Chapter 5

Building of the Working Model

5.1 Overview to Chapter

The aim of this chapter is to explain in detail the development of a working computer model which will be used to test the search algorithm on a number of real life examples (the testing is the subject of Chapter 6). This chapter will address the following step of the research method:

Develop a search model that will integrate a conventional search engine with concepts and methods (the algorithm) that will undertake the search. This would need to be a working model that can be tested using the Web.

Recall the overall algorithm concept presented at the end of Chapter 3 (Figure 5.1):

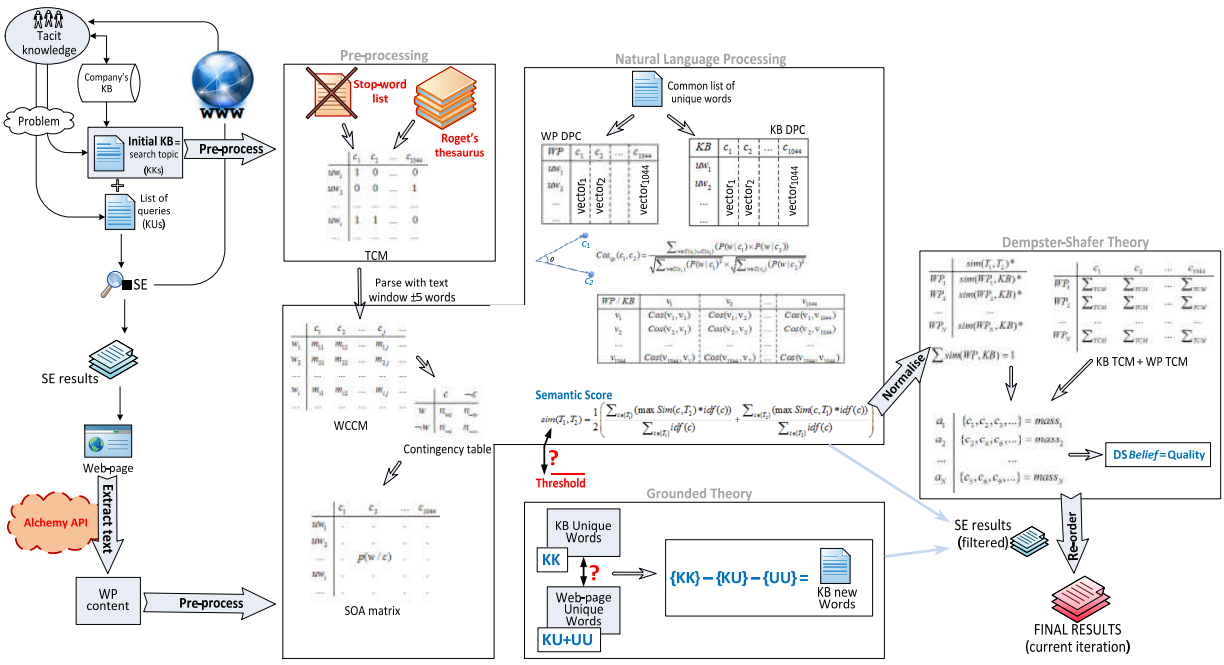


Figure 5.1 – Schematic Representation of the Search Model

The chapter is structured to provide a description of the model concept (section 5.2); detail of the model design (section 5.3); a system's design to facilitate the description of the software modules (section 5.4); and the full model software development (section 5.4). Section 5.4 is divided into Web-search and processing of Google results (section 5.4.1), pre-processing of Web-search results and semantic closeness measurement (section 5.4.2), Grounded Theory module (section 5.4.3) and the Dempster-Shafer module (section 5.4.4).

The working prototype was implemented using programming language Python. Full program code is available in the Appendix 4.

5.2 Model concept

Chapter 3 introduced the key elements of the proposed research algorithm. This section will describe key parts of the working model that will test the algorithm. The fundamental elements required for the algorithm to work are listed below.

- Initial knowledge base. This contains all information currently available for analysts on the selected search topic and summarised in the form of unstructured text, i.e. natural language. The size of the text file may vary and depends solely on the amount of knowledge on the search topic. Initial knowledge base is created via manual analysis and is recorded in a separate text files.
- List of queries for the search engine (Google is being used for this research)**.** **S**earch queries reflect the known unknowns about the search topic and define the search direction. Similarly to the initial knowledge base, queries are manually identified. Each query is run through the Google search and corresponds to a single iteration, i.e. a single complete system run.
- Roget's Thesaurus – thesaurus of English words where all words are categorised into 1,044 groups based on the word meaning. Such categories represent different semantic concepts and each English word can be associated with multiple concepts. The thesaurus is stored in a separate text file; the file size is 2.4MB. The thesaurus text file lists English words with the corresponding categories in both textual and numerical format. Below is an example fragment (Figure 5.2) of the thesaurus file:

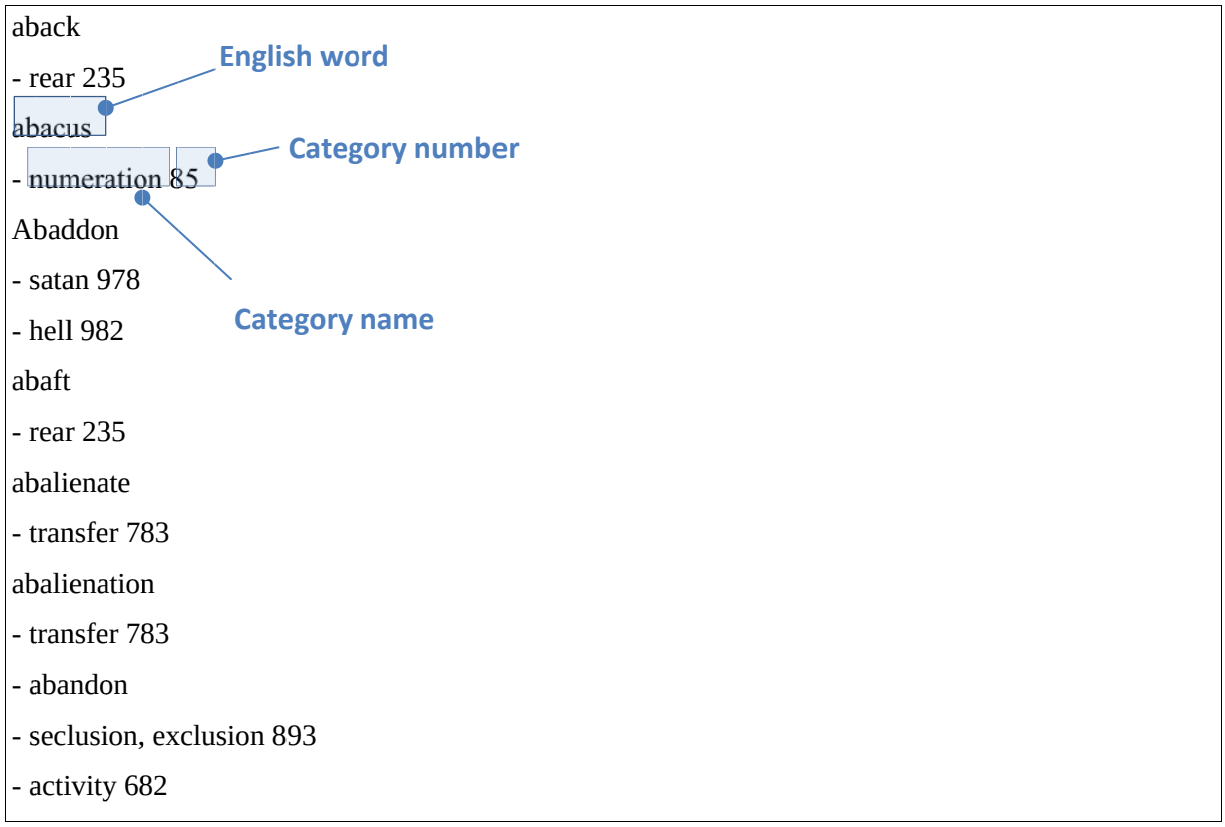


Figure 5.2 – Example of the Thesaurus file

- Stop-word list. This list contains words that are used frequently in the language, e.g. articles, pronouns etc. The list of stop-words is stored as a separate text file; the file size is 2.7KB. The standard list of 250 words (van Rijsbergen, 1979) has been extended for the purpose of this research – there are total of 369 stop-words used, including articled (e.g. a, the), pronouns (e.g. I, you, that, something), numbers (e.g. two, second). The complete list of stop words is attached in the Appendix 2.
- Search engine – this research extensively uses Google search as it has the largest index of the Web. For this research it is not an issue to be overly inclusive during the first stages of the algorithm. False positives will be filtered out during the Natural Language Processing stage.
- Python libraries and script files. The new search method is evaluated through a semi-automated model realised with Python. The core software module pre-processes texts (initial knowledge base and Web-pages extracted text), calculates semantic distance between them, and calculates the KU conversion rate. Other

modules pre-process the thesaurus file, transforming it into a collection of words/phrases with the corresponding categories indexes. They are also used for evaluation of the Dempster-Shafer quality analysis, as well as calculation of distribution profiles of concepts.

The choice of Python to model software development was driven by its simplicity along with excellent functionality for processing linguistic data. It can be downloaded from http://www.python.org) and installers are available for all platforms. Python is an object-oriented computer language that permits data and algorithms to be encapsulated within software modules which can be re-used easily. Python is a dynamic language and permits the easy addition of attributes to objects, and dynamic typing of variables, facilitating rapid development. Python comes with an extensive standard library, including components for textual processing, numerical processing, and Web connectivity (used for this research).

A significant part of the developed software was realised with Natural Language Toolkit (NLTK) – a leading platform for building Python programs to work with human language data (www.nltk.org). As described in (Bird, et al., 2009) NLTK provides basic classes for representing data relevant to natural language processing; standard interfaces for performing tasks such as part-of-speech tagging, syntactic parsing, and text classification; and standard implementations for each task which can be combined to solve complex problems. Python AlchemyAPI library (www.alchemyapi.com) is used for extraction of the main textual content from Web-pages (ignoring side bars, menus etc.).

The following sections will provide further detail on the model design (section 5.3) and the software (section 5.4).

5.3 Model design

Recalling the logic of the developed search method from Chapter 3 (Figure 3.1), and combining it with the mathematical parameters developed in Chapter 4, the detailed schematic of the working model will be as presented in Figure 5.3. This model schematic contains the research algorithm.

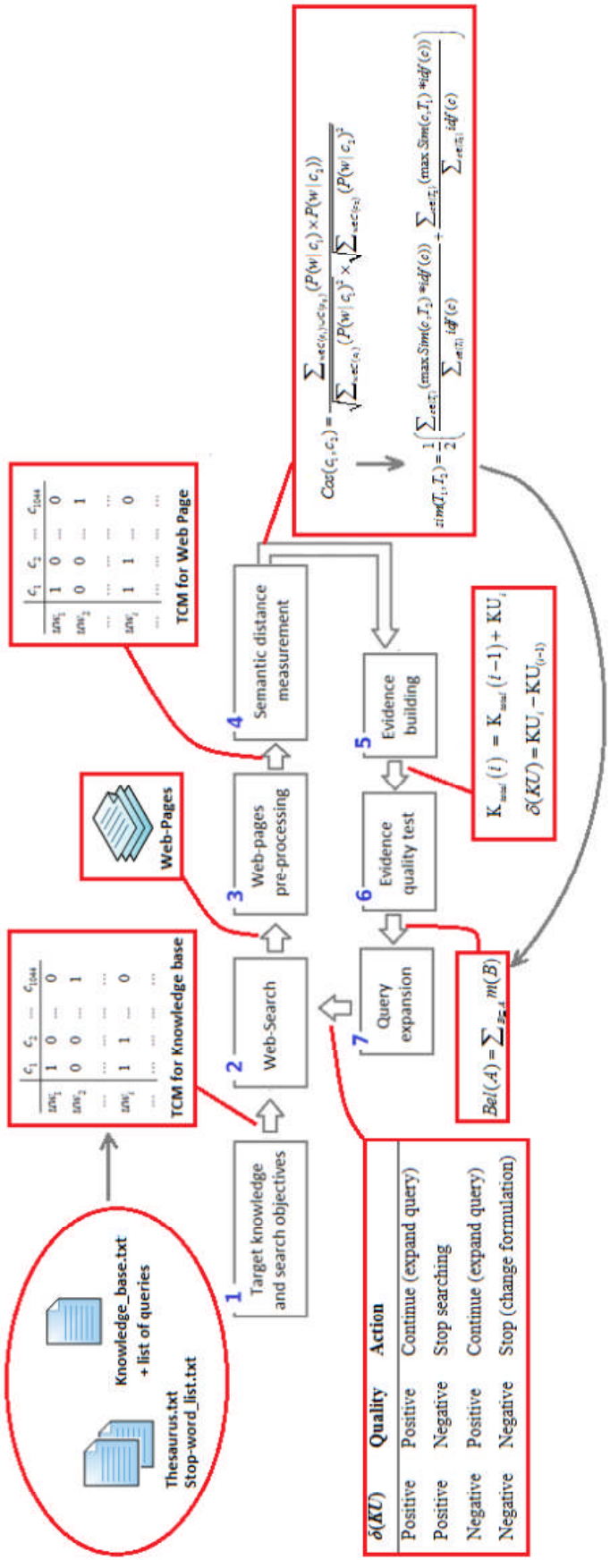


Figure 5.3 – A detailed schematic of the working model

The main module software elements are developed using the Unified modelling Language (UML); the class diagram is shown at Figure 5.4.

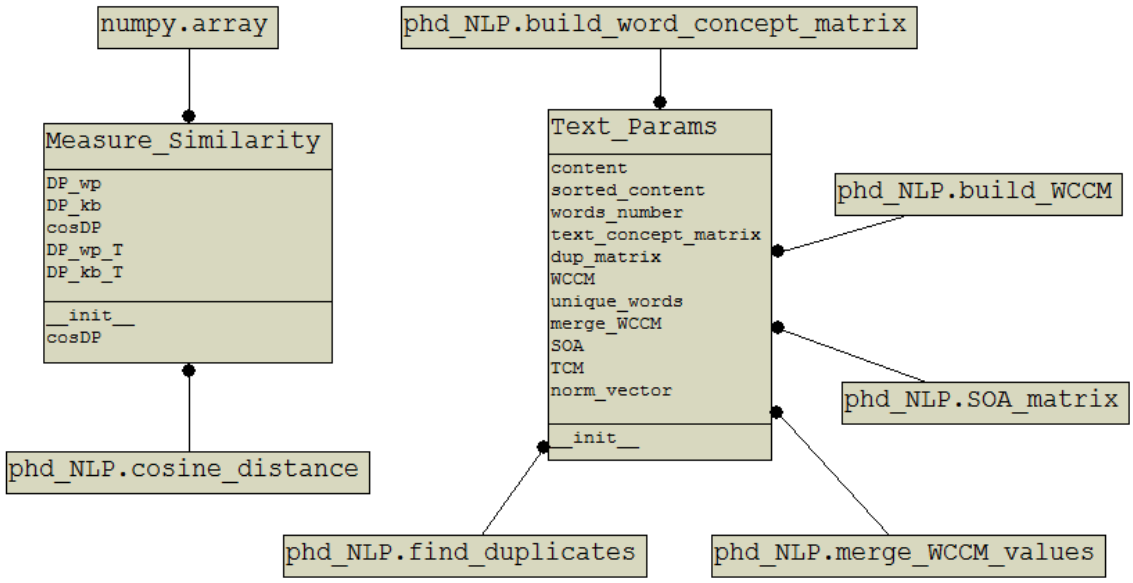


Figure 5.4 –UML class diagram for the main module

The UML sequence diagrams below show how system objects interact with each other in a given scenario. Figure 5.5 shows the UML diagram of the Main Module, figures 5.6 and 5.7 display UML diagrams for the Grounded Theory Module and Dempster-Shafer Module respectively.

Full size UML diagrams are attached in the Appendix 3 at the end of this thesis.

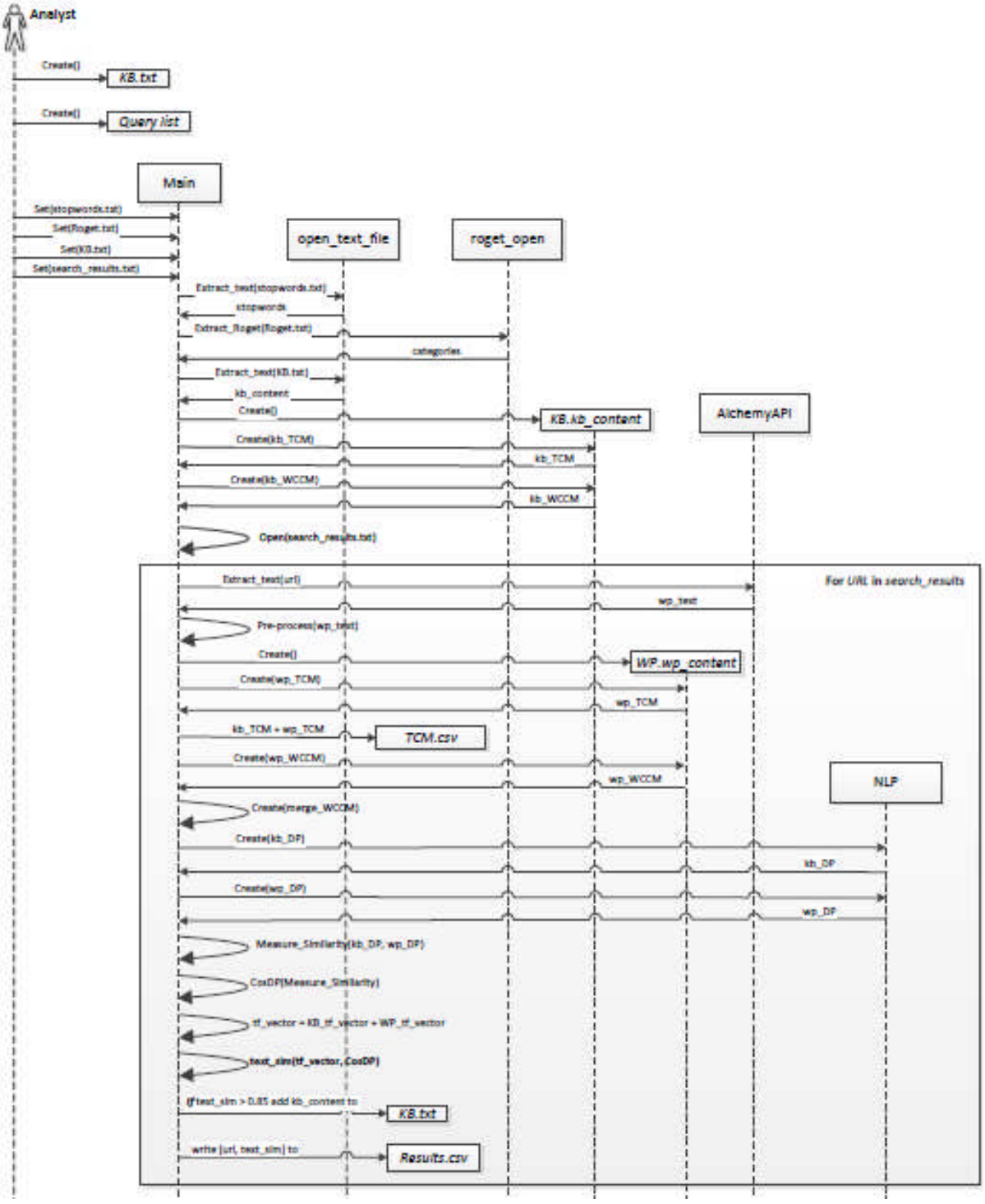


Figure 5.5 – UML diagram of the Main Module

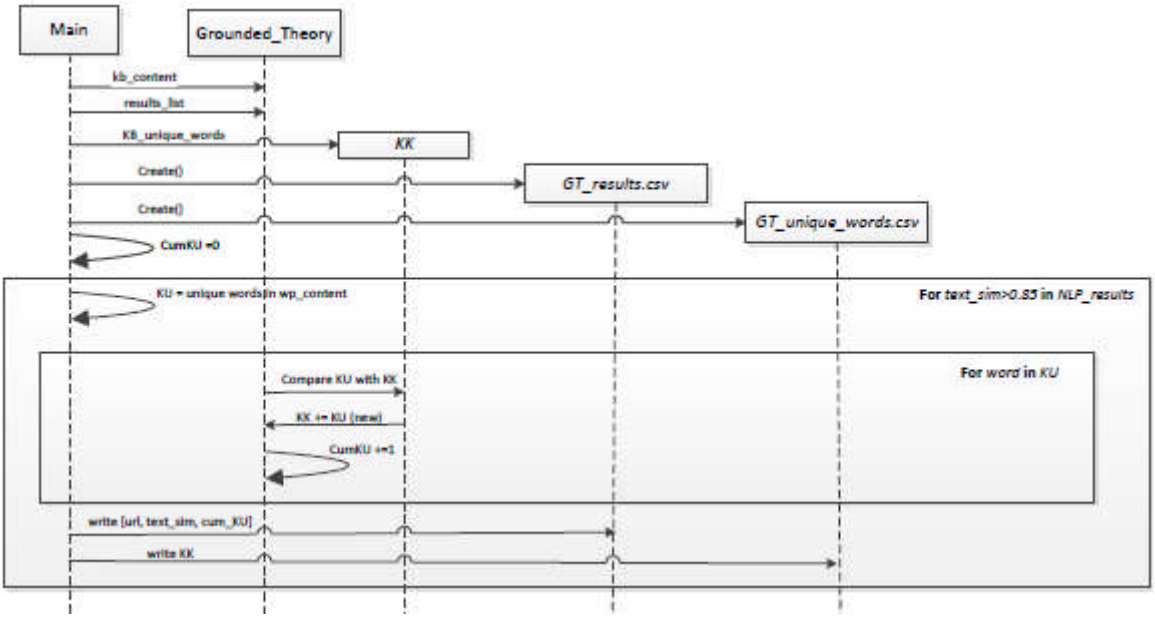


Figure 5.6 – UML diagram of the Grounded Theory Module

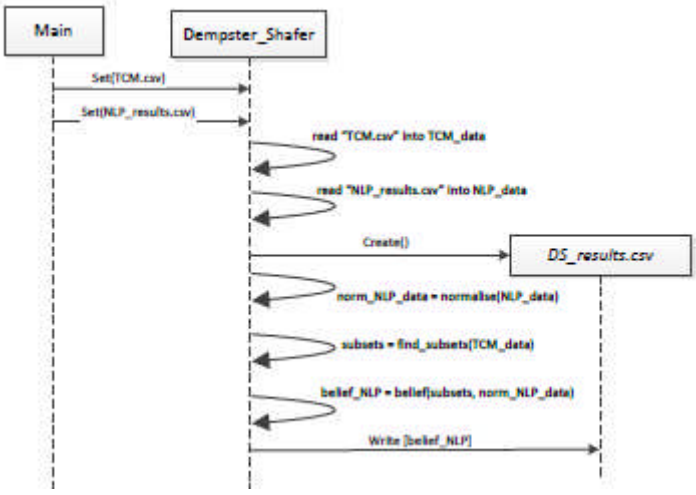


Figure 5.7 – UML diagram of the Dempster Shafer Module

5.4 Model software

The model was implemented using Python 3.2 programming language with MS Excel 2010 providing for the statistical analysis and visualisation of results.

5.4.1 Web-Search and processing of Google results

The first part of the algorithm is semi-automated owing to the difficulty of scraping of Google results since their algorithm and the source code is constantly being changed. Thus, for each query processed by Google, the source code for each of the returned results page was opened and stored in a same single text file. Then, the URL's were automatically extracted using the algorithm below.

Input: Google results source code *"Google_results.txt"*,

1 open and read file "*Google_results.txt*";
2 split text using space as a delimiter;
3 initialise an array of URL's as *results_list*;

4 for *URL* **in** *results_list*
5 **if** *URL* **starts with** *'href='*
6 **if** it contains *'Webcache.google'* OR *'.google.'*
7 remove *URL*;
8 *results_list* += *URL*;

9 write *results_list* to *"results_file.txt"*:

Output: *"results_file.txt"*

The resulting list of URL's to the query responsive Web-pages was stored as a text file and used an input in the next stage of the algorithm.

5.4.2 Pre-Processing and semantic closeness measurement

Next, the extracted text is set for the initial knowledge base. The identified Web-Pages are attempted to open and, if successful, the extracted text is set and processed for a Web-page following the *results_list* in the "*results_file.txt*". The semantic closeness score is then calculated between the knowledge base extracted text and each of the Web-pages returned by Google. NLTK and AlchemyAPI Python libraries are used in this stage.

The algorithm below demonstrates the logic behind calculating semantic distance between the initial knowledge base text and each Web-page text within a single iteration.

Input: knowledge base *"KB.txt"*,
Web-search results *"WP_list.txt"*,
list of stop-words *"stopwords.txt"*,
Roget's thesaurus *"thesaurus.txt"*,
AlchemyAPI,
NLTK

1 extract list of words and categories from "*thesaurus.txt*";
2 extract words from "*KB.txt*" using NLTK and exclude stop-words;
3 compute KB TCM;
4 compute KB WCCM;
5 compute KB SOA matrix;

6 initialise empty *"results.csv"*;

7 for *link* in *"WP_list.txt"*:
8 *WP* = open and read *link* using AlchemyAPI;
9 **if** *cannot open link* **then**
10 *text_similarity* = 0;
11 write *link* in 1st column and *text_similarity* in 2nd column of *"results.csv"*;
12 **else**
13 extract words from *WP* using NLTK and remove stop-words;

14 compute WP TCM;

15 compute WP WCCM;

16 compute WP SOA matrix;

17 compute DP of concepts for KB;

18 compute DP of concepts for WP;

19 compute a matrix with *Cos* values between each element from KB DP and WP DP;

20 compute TF vector with term frequencies values for each of 1044 concepts;

21 compute *text_similarity* as semantic distance;

22 write *link* in the 1st column and *text_similarity* in 2nd column of *"results.csv"*;

Output: *"results.csv"*

At this stage, before writing the *text_similarity* results to "*results.csv"*, the initial knowledge base may be expanded with Web-pages extracted text, if the *text_similarity* value for the two texts is greater than the chosen threshold of 0.85.

The output is a comma-separated value file containing the list of links to Web-Pages with their corresponding semantic closeness scores.

To explain further, below are the details of the functions used in the above algorithm.

1 extract list of words and categories from "*thesaurus.txt*";

Roget's Thesaurus file is read and recorded into array, which is then used as a look-up table to identify corresponding categories for used English words.

Input: Roget's thesaurus "*thesaurus.txt*",

NLTK

1 open and read file "*thesaurus.txt*";

2 initialise *thesaurus_array*

3 initialise *categories_buffer*

4 for *line* in *"thesaurus.txt"*:
5 append current *thesaurus_array* word with the corresponding *categories_buffer*;
6 **if not** *line* contains category reference
7 add *line* to *thesaurus_array*
8 empty *categories_buffer*
9 **else**
10 store category number in the *categories_buffer*

Output: *thesaurus_array*

2 extract words from "KB.txt" using NLTK and exclude stop-words;

First, the list of stop-words is processed into array. Next, the initial knowledge base is opened, text is extracted and single words are stored in array, removing the stop-words.

Input: list of stop-words "*stopwords.txt*",
initial knowledge base text "*KB.txt*",
NLTK

1 extract_text from "*stopwords.txt*" and write to *stopwords* array
2 extract_text from "*KB.txt*" and write to *KB_words* array
3 initialise array *KB_content*
4 for *word* in *KB_words*
5 **if not** *word* in *stopwords*
6 append *KB_content* with *word*

Output: *stopwords, KB_content*

***
extract_text (file)
1 open and read file "*file.txt*";

2 tokenise text with NLTK

3 record text tokens into array *text*

Return *text*

Initial Knowledge base text (*KB_content*) is then prepared for further processing. Please refer to Chapter 4 for full details on the mathematics behind.

3 compute KB TCM;

The Text-Concept Matrix (TCM) is built, the matrix is of the dimension [n_u x 1044], where n_u is the amount of unique words within the text. TCM values are either "1" indicating that the word is associated with the concept in the thesaurus, or "0" in the other case.

Input: initial knowledge base text *KB_content*

NLTK

1 build_TCM for *KB_content*

Output: *KB_TCM*

***

build_TCM(*text_content*, *thesaurus_array*)

1 *UW* = number of unique words in *text_content*;

2 initialise zero *TCM* matrix [1044 x *UW*];

3 for *word* in *text_content*

4 **if** *word* is in *thesaurus_array*

5 get list of corresponding categories numbers

6 increment *TCM[word, category]* by "1"

Return *TCM*

4 compute KB WCCM;

Based on the TCM, a Word-Category Co-occurrence Matrix (WCCM) is built. A matrix cell m_{ij}, corresponding to word w_i and concept c_j, is populated with the number of times w_i co-occurs with any word listed under concept c_j in the thesaurus within a text window of ±5 words. Ad discussed in Chapter 4, the matrix is of the type:

	c_1	c_2	...	c_j	...
w_1	m_{11}	m_{12}	...	m_{1j}	...
w_2	m_{21}	m_{22}	...	m_{2j}	...
...	...	...	...	...	...
w_i	m_{i1}	m_{i2}	...	m_{ij}	...
...	...	...	...	...	...

Input: Text-Concept Matrix *KB_TCM*

NLTK

1 build_WCCM for *KB_TCM*

Output: *KB_WCCM*

***

build_WCCM(*TCM*)

1 initialise zero *WCCM* matrix [*TCM_rows* x 1044];

2 for *word* in *TCM[word]* *#matrix row*

3 **for** (*i=word-5)* ***to*** *(word+5)*

4 **if not** *i== word*

5 *WCCM[word]+= TCM[word]* *# sum up all rows within text window of ±5 words*

Return *WCCM*

Thus, for every word in the text, values for its neighbouring 10 words within the text window are found in the TCM. The corresponding TCM rows are added up and the resulting row is added to the new matrix WCCM. The resulting matrix is of the size [N x 1044], where N is

the number of all words in the texts excluding stop-words. Each value C_{ij} corresponds to a number of times a word *i* co-occurred with any word listed under the category *j* within the text window of ±5 words.

5 compute KB SOA matrix;

The next step is creation of the SOA matrix which elements correspond to values of conditional probabilities of a word w co-occurs with other words associated with a concept c. SOA matrix can be seen as a collection of distributional profiles of concepts for a given text.

Input: *KB_WCCM*

NLTK

1 *KB_merge_WCCM* = **merge_WCCM_values** for *KB_WCCM*

2 build_ SOA_matrix for *KB_merge_WCCM*

Output: *KB_SOA_matrix*

***

merge_WCCM_values (*WCCM, UW*)

1 initialise zero *merge_WCCM* matrix [*UW* x *1044*]; *# UW – number of unique words*

2 for *word* in *WCCM* *#matrix row*

3 *add_WCCM* = add up all rows for *WCCM[word]*

4 *merge_WCCM [word] = add_WCCM[word] # sum up all rows for the same word*

Return *merge_WCCM*

build_SOA_matrix (*merge_WCCM, UW*)

1 initialise zero *SOA* matrix [*UW* x *1044*];

2 for *row* in *merge_WCCM* *#for all matrix rows*

3 **for** *col* in *row* *#for all column elements in the row*

4 **if** *col* == 0

```
5          SOA[row, col] = 0
6        else
             # SOA as conditional probability p(w|c)
7          SOA[row, col] = merge_WCCM[row, col] / sum(merge_WCCM[ : , col])
Return SOA
```

```
6 initialize empty "results.csv";
```

The results will be recorded in the "results.csv" file from the array "*results*" in the format [URL, Semantic closeness value] for each URL in the list of search engine results, following the logic below.

```
7 for link in "WP_list.txt":
 8    WP = open and read link using AlchemyAPI;
 …
13       extract words from WP using NLTK and remove stop-words;
14       compute WP TCM;
15       compute WP WCCM;
16       compute WP SOA matrix;
```

At this stage the system will attempt to open Web-pages from the list of search engine results and, if it successfully gets extracted text, it will follow the same logic as with processing of the initial knowledge base text.

```
Input: Web-search results "WP_list.txt",
       list of stop-words "stopwords.txt",
       Roget's thesaurus "thesaurus.txt",
       Empty results file "results.csv"
       AlchemyAPI,
       NLTK
```

1 open and read file "*WP_list.txt*";

2 extract *URL*'s from "*WP_list.txt*" using NLTK;

3 initialise array *WP_content;*

4 for *URL* **in** "*WP_list*":

5 **try**:

6 *wp_text* = **Get_URL_Text(*URL*)** using AlchemyAPI;

7 **except:**

8 *results*[*URL*] = [0,0]

9 write *results*[*URL*] to "*results.csv*"

similarly to processing of *KB_content*

10 *WP_content* = **pre_process_wp_text**(*wp_text*) # *remove stop-words and prefix*

11 *WP_TCM* = **build_TCM** for *WP_content*

12 *WP_WCCM* = **build_WCCM** for *WP_TCM*

13 *WP_merge_WCCM* = **merge_WCCM_values** for *WP_WCCM*

14 *WP_SOA* = **build_ SOA_matrix** for *KB_merge_WCCM*

Output: *WP_SOA*

17 compute DP of concepts for KB;

18 compute DP of concepts for WP;

Distributional profiles of concepts are computed for concepts of the initial knowledge base and the Web-page extracted text currently being processed.

Input: *KB_SOA, WP_SOA*

1 *common_unique_words* = augmented array of *KB_unique_words* and *WP_unique_words*;

2 *len_DP* = length of *common_unique_words*;

3 *DP_KB* = **create_DP**(*common_unique_words, len_DP, KB_unique_words, KB_SOA*)

4 *DP_WP* = **create_DP**(*common_unique_words, len_DP, WP_unique_words, WP_SOA*)

Output: *DP_KB, DP_WP*

***

create_DP(***common_unique_words, len_DP, unique_words, SOA***)

1 initialise zero *DP* matrix [*len_DP* x *1044*];

2 for *word* in *common_unique_words*

3 **if** *word* **in** *unique_words* *# if the word is present in the current text*

4 *DP*[*word*] += *SOA*[*word*] *# add the entire SOA row for co-occurred words*

5 *SOA*[*row, col*] = 0

Return *DP*

19 compute a matrix with *Cos* values between each element from KB DP and WP DP;

At this stage, semantic closeness is calculated between every concept in the initial knowledge base and every concept in the Web-page's extracted text.

Input: *DP_KB, DP_WP*

1 *cosDP* = **cosDP**(*DP_KB, DP_WP*) *#build matrix of Cosines between DPs of concepts*

Output: *cosDP*

***

cosDP (***DP_KB, DP_WP***)

1 initialise zero cosDP matrix [1044 x 1044];

2 *DP_WP_T = DP_WP* transposed; # Transpose both DP matrices for further processing

3 *DP_KB_T = DP_KB* transposed;

4 initialise zero cosDP matrix [1044 x 1044];

5 initialise zero cosDP matrix [1044 x 1044];

6 for *WP_row* **in** *DP_WP_T*

```
7    for KB_row in KB_WP_T
8        # fill matrix with Cosine values between concept vectors
9        cosDP[WP_row, KB_row] = cosine_distance(WP_row, KB_row)
Return: cosDP


cosine_distance(u, v)
  # Cosine of the angle between vectors v and u. This is equal to Cos = u·v / |u||v|
1 cd = dot_product(u, v) / (sqrt (dot_product (u, u)) * sqrt (dot_product (v, v)))
Return: cd
```

```
20    compute TF vector with term frequencies values for each of 1044 concepts;
```

Function that creates a vector with a term frequency value for each of the 1044 concepts. It will be used as a weighting parameter when calculating the final semantic closeness score.

```
Input: KB_TCM, WP_TCM


1 idf = idf_matrix(WP_TCM, KB_TCM)        # create idf values matrix [1044 x 1044]
2 KB_tf_vector = fill_tf_vector(KB_merge_WCCM))
3 WP_tf_vector = fill_tf_vector(WP_merge_WCCM))
4 tf_vector = KB_tf_vector + WP_tf_vector


Output: tf_vector
***
fill_tf_vector(merge_WCCM)
1 merge_WCCM_T = merge_WCCM transposed;   # to be able to add values for concept by
row
2 initialise empty array tf_vector;
3 for i in 1044:
4    concept_sum = sum(merge_WCCM_T [ i ] )
5    tf_vector += concept_sum
```

Return: *tf_vector*

21 compute *text_similarity* as semantic distance;

Next function calculates semantic closeness value. As was explained in Chapter 4, the similarity between two texts T_1 and T_2 is determined using the following function:

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{c \in \{T_1\}} (\max Sim(c, T_2) * idf(c))}{\sum_{c \in \{T_1\}} idf(c)} + \frac{\sum_{c \in \{T_2\}} (\max Sim(c, T_1) * idf(c))}{\sum_{c \in \{T_2\}} idf(c)} \right).$$

Input: *tf_vector, cosDP*

1 *text_sim* = **text_similarity***(tf_vector, cosDP)*

Output: *text_sim*

***

text_similarity(tf_vector, cosDP)

1 *left_num* = 0; *# left numerator – WP concepts to KB concepts*

2 *right_num* = 0; *# right numerator – KB concepts to WP concepts*

3 *tf_sum* = 0; *#denominator*

 # calculating left part of the equation – WP concepts to KB concepts

4 for *row* **in** *cosDP*

5 *left_num* += *row.max*() * *tf_vector*[*row*]; *# numerator*

6 *tf_sum* += *tf_vector*[*row*]; *# denominator*

7 *wp_to_kb* = *left_num* / *tf_sum*;

 # calculating right part of the equation – KB concepts to WP concepts

4 for *row* **in** *cosDP_T* *# transposed cosDP matrix*

5 *right_num* += *row.max*() * *tf_vector*[*row*] *# numerator*

6 *tf_sum* += *tf_vector*[*row*] *# denominator*

7 *kb_to_wp* = *right_num* / *tf_sum*

 # TOTAL similarity score between two texts

8 *text_sim* = (*wp_to_kb* + *kb_to_wp*)/*2* *# as per the formula*

Return: *text_sim*

22 write *link* in the 1st column and *text_similarity* in 2nd column of *"results.csv"*;

This file will be then used to analyse the results for quality. There is one more column to be written in the "*results.csv*" – the amount of discovered UU and KU. Next paragraph will explain how the Grounded Theory module is implemented.

5.4.3 Grounded Theory module

Having obtained semantic similarity values, the Grounded Theory based completeness test is then run to identify previously unknown data and calculate the conversion rate of KUs.

It is assumed, that the initial knowledge base is the only information known on the search topic. Thus, it is treated as the collection of KKs. Within a single iteration, every new Web-page extracted text that has the semantic closeness score over 0.85 is analysed for the amount of new concepts, compared to the KK collection.

As discussed in Chapter 4, following each iteration, newly identified concepts are added to the initial knowledge base, thus expanding the evidence:

$$\mathrm{K}_{total}(i) = \mathrm{K}_{total}(i-1) + \mathrm{KU}_i,$$

where KU represent new concepts on each iteration, and KU(i–1) ≠ KU(i). A change in KU represents the conversion rate $\delta(KU)$ of new concepts in evidence and is defined as:

$$\delta(KU) = \mathrm{KU}_i - \mathrm{KU}_{(i-1)}.$$

The KU conversion rate is evaluated using the following algorithm:

```
Input: kb_content, results_list

1 KK = UW_kb_content;          # list of Known Knowns – unique words in KB text
2 initialise and open "GT_unique_words.csv"; # global list of unique concepts
3 initialise and open "GT_results.csv";      # current list of unique concepts
4 cum_KU = 0;                  # cumulative amount of KUs for KU conversion rate


5 for URL in results_list
6    calculate semantic similarity text_sim;
7    if text_sim > 0.85
8       for UW in UW_wp_content     # unique words in the Web-pages extracted text
9          if UW not in KK
10            KK += UW;           # add current word to the list of KKs
11            cum_KU +=1;

12       write [URL, text_sim, cum_KU] to "GT_results.csv";   # GT results for current query
13       write KK to "GT_unique_words.csv"                    # global list of KKs

Output: "GT_results.csv", "GT_unique_words.csv"
```

5.4.4 Dempster-Shafer module

The final stage is the quality test of the discovered information using the Dempster-Shafer theory. The total quality score for each Web-page is calculated as a Belief function; the higher the value of Belief for a Web-page, the higher its quality value and the higher the Web-page will be ranked in the final list of results.

The algorithm below demonstrates the logic behind calculating the Quality score for a search result. Two "csv" files are required for input:

- True/False values for concept appearance in the documents;
- Matrix of mass functions for each source of evidence.

```
Input: "TCM.csv",          # True/False values for concept appearance
       "NLP_results.csv"   # matrix of mass functions

1 open and read "TCM.csv" into TCM_data;
2 open and read "NLP_results.csv" into NLP_data;

  # calculate Belief function
3 norm_NLP_data = normalise(NLP_data);          # mass functions
4 subsets = find_subsets(TCM_data);             # returns groups of subsets of every row
5 belief_NLP = belief(subsets, norm_NLP_data);

7 DS_results = [belief_NLP];
8 write results DS_results to "DS_results.csv"

Output: "DS_results.csv"
```

To explain further, below are the details of the functions used in the above algorithm.

```
1 open and read "TCM.csv" into TCM_data;
2 open and read "NLP_results.csv" into NLP_data;
```

The first file corresponds to the Text Concept Matrix of the size [N x 1044], where columns correspond to 1044 categories in Roget's Thesaurus, and N rows represent URL's of Web-pages that successfully passed the semantic closeness test. If a thesaurus concept ***c*** is used in the extracted text ***n***, the corresponding value *TCM*[***n***,***c***] = 1, otherwise 0. The second file contains a list of the semantic closeness values (*text_sim*) for the successful Web-pages within a single iteration.

3 *norm_NLP_data* = normalise(*NLP_data*); *# mass functions*

The mass function for a text, as a proportion of all evidence that supports the text, is a normalised value of the NLP analysis result according to $sim(T_1, T_2)$. Recalling from Chapter 4, the calculated semantic distance has a value between 0 and 1, and the total evidence is scaled to fall between 0 and 1 in order to satisfy the definition of *bpa*:

$$m(\varnothing) = 0 \text{ and } \sum_{A \subseteq U} m(A) = 1.$$

4 *subsets* = **find_subsets**(*TCM_data*); *# returns groups of subsets of every row*

This function checks if any of the vectors in TCM is a subset of another TCM vector. It creates a list of the type [TCM URL index, subset index, subset index, …] and returns groups of subsets of every vector within the same "*TCM.csv*".

find_subsets(array)

1 initialise zero array *subsets;*

2 *array_copy = array;*

3 for *row* **in** *array*

4 *subset_rows*[*row*] = all subsets of *row* in *array_copy;*

5 *subsets*[*row*] += *subset_rows*[*row*]

Return *subsets*

Having obtained the list of all subsets, belief function can be then calculated.

5 *belief_NLP* = **belief**(*subsets*, *norm_NLP_data*);

The quality of information in each text is associated with the value of the Belief function for corresponding texts and reflects the amount of evidence supporting each text directly. Each text can be seen as a combination of Thesaurus concepts. Following from Chapter 4, the

probability values are assigned to sets of possibilities and Belief in a hypothesis is constituted by the sum of the masses of all sets enclosed by it:

$$Bel(A) = \sum_{B \subseteq A} m(B) .$$

```
belief(subsets,NLP_data)
1 initialise empty array belief;
2 for row in subsets      # row is of the type [TCM index, subset_1 index, subset_2 index, …]
3    bel = 0;
4      for index in row
          # add up "text_sim" values for the current text and all its TCM subsets
5          bel += NLP_data[index];
6    belief += bel
Return belief
```

Results are then written to the csv file "*DS_results.csv*".

Based on the results of tests for completeness in "*GT_results.csv"* and the quality test results in "*DS_results.csv*", a decision is made on whether to iterate or stop.

5.5 Summary

This chapter provided a comprehensive overview to the model design and software. Fundamental elements required for the algorithm to work were discussed. Model design was described with UML class diagram for the main module and described the structure of the system by showing the system's classes, their attributes, methods, and the relationships among objects. The UML sequence diagrams for all three modules of software showed how the system objects interact with each other in time and using a given scenario.

Pseudo-code was used to describe key pieces of code corresponding to all stages of the algorithm. As such, description was provided to Web-Search and processing of Google results, pre-processing of extracted text and semantic closeness measurement stage,

Grounded theory based completeness measurement, as well as measurement of quality score for semantic related content.

Chapter 6

Experiments

6.1 Overview to Chapter

Following the development of model design and software discussed in the previous chapter, this chapter specifically addresses the following step of the research method:

> *Test the model through use cases and undertake analysis of the results. Develop experiments that would address the efficiency of the model to find the Web content that is relevant to the defined search topic, as well as to track the discovery rate of unknown unknowns, and to provide a reliable quality metric that would assess the discovered information from the richness viewpoint.*

This chapter provides 'use case' specifications and associated data collection (section 6.2), model testing to assess the efficacy of the research-developed algorithm (section 6.3), experimental runs against use cases together with analysis (section 6.4), and consolidation of results (section 6.5)

The aim of the experiments is to evaluate whether the research method significantly improves the process of decision making by extracting only relevant high quality information from the Web. The evaluation experiments cover a predefined set of chosen test topics with corresponding lists of queries.

Recalling the objective of this research is to develop an algorithm that facilitates topic related search for Web-based information and estimates the quality of the extracted content, proving it is significantly enhanced comparing to the traditional Web-search approaches. Experiments are planned and conducted in order to provide assessment of the built system addressing each aspect of the research objective assuming varying level of competency in the search topic.

The first part of the experiments was aimed at assessing how well the system can find topic-related quality information from the Web when the topic is well understood and well defined. The second part of the experiments was aimed at assessing the ability of the

system to identify relevant information of high quality assuming that the topic understanding is incomplete and the queries were partially irrelevant to the search topic. Both parts use the same knowledge base for each iteration of the model, and the knowledge base that is appended every time if a successful (relevant) extracted text is identified.

For each experiment the conversion rate of unknowns was taken as a parameter for identifying when the iterating should stop. The cumulative sum of discovered unknowns was analysed and, if the next iteration has a little conversion rate, it is assumed that the topic is close to be exhausted and new iteration on the same topic will not significantly enhance the knowledge base.

Five experiments have been devised for this research and described in section 6.4:

- Experiment 1 and 2 – Same well-defined knowledge base (6.4.1 and 6.4.2)
- Experiment 3 – Appended well-defined knowledge base (6.4.3)
- Experiment 4 – Same poorly-defined knowledge base (6.4.4)
- Experiment 5 – Appended poorly-defined knowledge base (6.4.5)

The experiment results are illustrated with various graphs and tables.

6.2 Use-case Specifications and Data Collection

The experimental part covers three chosen test topics with corresponding lists of queries. A working prototype was created and implemented using Python (see Chapter 5 for details). The experimental element of this research is semi-automated – the initial knowledge base and the corresponding list of queries for a search engine are manually defined. In addition, some algorithm stages are manually processed.

Two options were considered while developing experiments to test the performance of the working model:

- within a single iteration, content of each new Web-page is compared separately against the same knowledge base initiated at the outset;
- the initial knowledge base is appended by the Web-page extracted text if it was considered relevant after the Natural Language Processing stage, thus, the next Web-page content is compared against the extended knowledge base.

For the purpose of this research, three search topics were chosen – “tobacco industry” (full text length 308 words), “cocaine smuggling” (full text length 477 words), and

"coffee production" (full text length 3,068 words). Initial knowledge bases were created via manual analysis and are recorded in form of unstructured text in separate text files. Full text of the three initial knowledge bases can be found in the Appendix at the end of this thesis.

Experiments are split in two parts. The first part of each experiment (Experiments 1, 2 and 3) has the objective of assessing the system's ability to find and filter Web-pages given a well-defined initial knowledge base and relevant queries. Two topics were used – "tobacco industry" and "cocaine smuggling", queries were chosen to be semantically focused around the search topic.

The second part of the experiment (Experiments 4 and 5) is with assessing the ability of the system to identify relevant information of high quality for a search topic, assuming that the topic understanding is incomplete and poorly defined. Search topics used ware "coffee production" and "tobacco industry".

For the topic "tobacco industry" the corresponding list of queries consisted of Q1 "tobacco industry competition", Q2 "tobacco criminal", Q3 "tobacco financial market", Q4 "tobacco smoking statistics", Q5 "tobacco investment opportunities".

The queries for the search topic "cocaine smuggling" are Q1 "cocaine production and distribution", Q2 "cocaine trafficking UK", Q3 "cocaine smuggling UK", Q4 "cocaine cartels", Q5 "Mexican drug cartels".

For the topic "coffee production" the corresponding set of associated queries included Q1 "coffee industry", Q2 "coffee criminal" and Q3 "coffee financial market".

In the second part of the experiments, when "coffee production" is the topic of the initial knowledge base (Experiment 4), queries used are associated with "tobacco industry" search topic". On the other hand, queries associated with "coffee production" search topic were used in conjunction with "tobacco industry" initial knowledge base (Experiment 5). In other words, the chosen queries were semantically remote from the search topic.

Initial knowledge base for the search topics "tobacco industry", "cocaine smuggling" and "coffee production" are attached in the Appendix 1.

Following the algorithm, the Web-search step was performed and the above queries for the three search topics were run through the Google search engine. Table 6.1 summarises Google search results on those queries (as of August 2013).

Query	Google Search Results
Search topic – "tobacco industry"	
"tobacco industry competition"	715
"tobacco criminal"	413
"tobacco financial market"	455
"tobacco smoking statistics"	855
"tobacco investment opportunities"	127
Total	**2,565**
Search topic – "cocaine smuggling"	
"cocaine production and distribution"	552
"cocaine trafficking UK"	631
"cocaine smuggling UK"	555
"cocaine cartels"	739
"Mexican drug cartels"	426
Total	**2,903**
Search topic – "coffee production"	
"coffee industry"	347
"coffee criminal"	261
"coffee financial market"	580
Total	**1,188**

Table 6.1 - Google search results (as of August 2013)

Thus, for each query processed by Google, the source code for each of the returned results page was opened and stored in a same single text file. The URL's were automatically extracted using the algorithm described in Chapter 5, Section 5.4.1. The collected list of responsive Web-pages for each query is an input for the next step of the algorithm (see Chapter 5, Section 5.4.2) where the main textual content is extracted and pre-processed.

6.3 Model Testing

Model testing was performed by comparing results of small logical parts of code with the manually evaluated results of calculations and analysis. Testing was done using a knowledge base "cocaine smuggling" (see Appendix 1 for full text). The working model was set up to compare textual content of each new Web-page separately against the same

knowledge base initiated, i.e. not appending the existing knowledge base with successful texts.

The query “cocaine production and distribution” was chosen to be focused around the search topic. Out of 552 Google results (see Table 6.1) a 10% random sample was chosen from the list of URLs returned by the search engine on the query. Thus, a sample of 55 Web-pages were randomly chosen and manually reviewed, identifying the textual content of the Web-pages as either “Relevant” or “Not Relevant”. The same sample of 55 Web-pages was processed using the new algorithm. The manual human coding results were then compared against the results of the working model.

As has been stated in Chapter 3, Section 3.5, the algorithm considers any extracted text as “Relevant” if its semantic closeness score is over 0.85. Table 6.2 demonstrates the coding results for the random sample of Web-pages responsive to the test search query.

Sample Web-Pages	**System Coding**	**Human Coding**	**Same Coding**	**Errors**
55	Relevant – 45 Not Relevant – 10	Relevant – 32 Not Relevant – 23	30	False Positives – 15 False Negatives – 2

Table 6.2 – Random sample relevance coding results

The full list of testing results is available in the Appendix 5 at the end of this thesis.

Precision and recall metrics were used to assess the accuracy of the analysis. The results below summarises the testing results.

$$\text{Precision} = \frac{\{\text{TruePositives}\}}{\{\text{TruePositives}\} + \{\text{FalsePositives}\}} = \frac{30}{30+15} = 67\% \qquad (6.1)$$

$$\text{Recall} = \frac{\{\text{TruePositives}\}}{\{\text{TruePositives}\} + \{\text{FalseNegatives}\}} = \frac{30}{30+2} = 93\% \qquad (6.2)$$

Thus, in the selected random sample of Web-Pages, the system was able to identify 93% of the relevant content recognised by manual analysis. Precision rate is lower and is estimated as 67%. This is due to the Web-pages that contain large amount of textual content, only some of which is topic-related, however it was picked up by the system as Relevant. In

this case, human relevance tagging would be considering the Web-Page to be rather irrelevant.

6.4 Model Runs

6.4.1 Experiment 1 – Same well-defined knowledge base

To assess the system's ability to find and filter Web-pages given a well-defined initial knowledge base and relevant queries, two experiments were run. The search topics were "tobacco industry" and "cocaine smuggling". These experiments are aimed to test the performance of the system when each new Web-page content is compared against the same knowledge base initiated at the beginning.

In the first experiment the topic of the initial knowledge base is "tobacco industry" and the chosen queries were Q1 "tobacco industry competition", Q2 "tobacco criminal", Q3 "tobacco financial market", Q4 "tobacco smoking statistics", Q5 "tobacco investment opportunities". The chosen queries are semantically focused around the search topic.

This experiment resulted in the significant reduction of information provided by Google search across all five queries. Q1 had 715 results in the search results list, 572 of them were considered relevant and 68 were classified as good quality Web-pages with 3 "high-quality" pages. The search results list for Q2 had 413 Web-pages, 345 pages were selected after the Semantic Closeness stage, 19 pages contained quality information with 5 pages considered as "high-quality". Q3 had 455 links to Web-pages in the search engine list, 248 were relevant, 18 were considered as containing quality information and no Web-pages were identified as containing "high-quality" content. The list for Q4 consisted of 855 links, 650 of them were selected after the Semantic Closeness stage, 57 were considered as quality information with 8 Web-pages being recognised as "high-quality". Q5 had 127 Web-pages in the search engine list, 93 passed through the semantic relevance test, 31 of them were considered to be quality information with only 1 page classified as "high quality".

Figure 6.1. shows reduction in Web-pages amount after three stages of filtering is shown in, distribution of semantic closeness values across five queries is shown in figure 6.2.

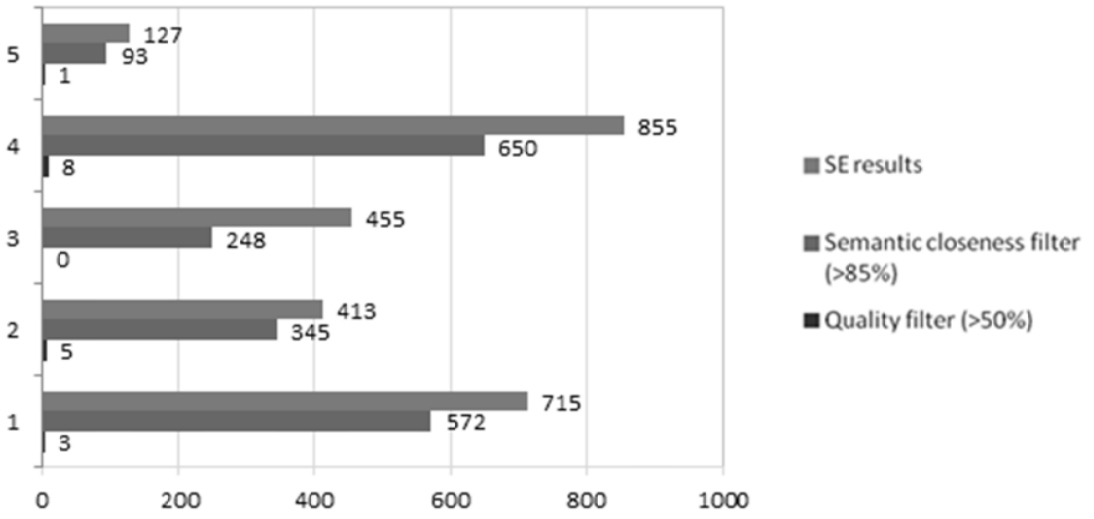


Figure 6.1 –Reduction in Web-pages amount after three stages of filtering (Experiment 1)

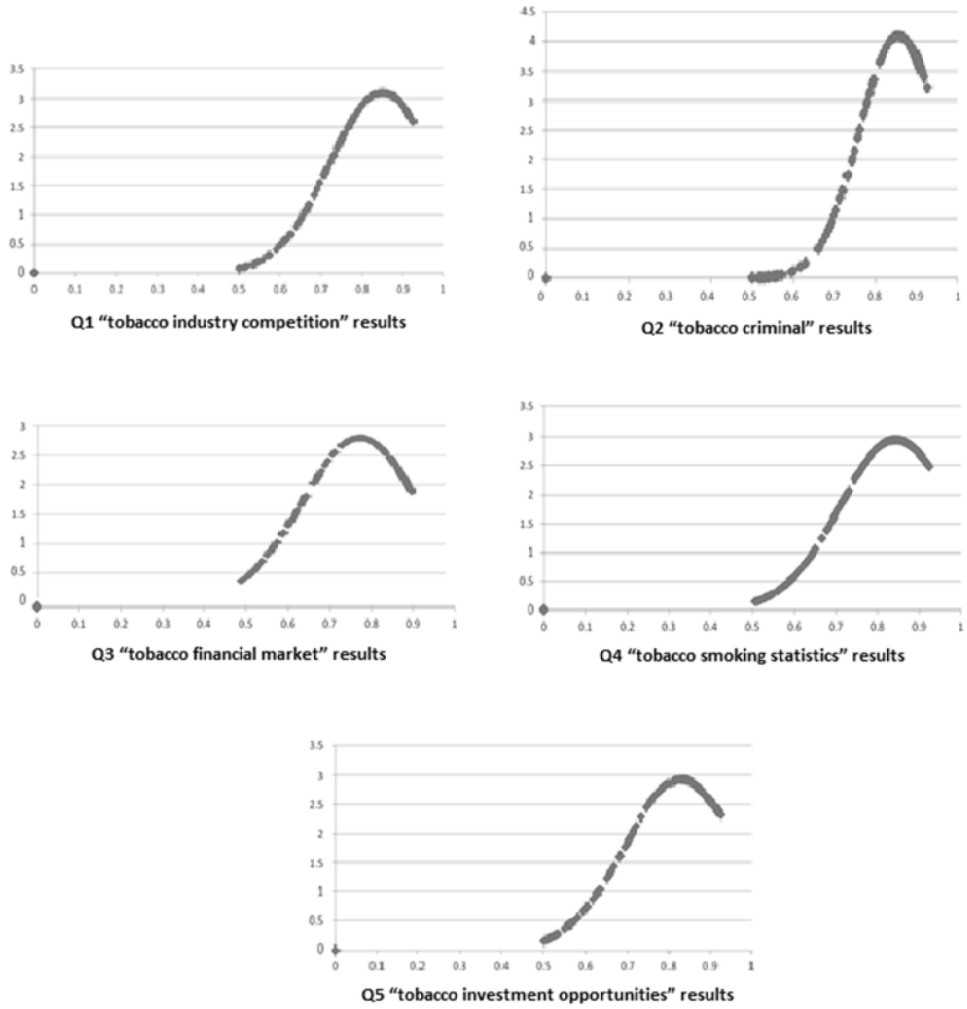


Figure 6.2 – Distribution of semantic closeness values across five queries (Experiment 1)

Using the same initial knowledge base for the search topic "tobacco industry" in this experiment resulted in the significant reduction of information provided by Google search across the five queries. Out of total of 2,565 Web-pages suggested by the search engine, 75% were recognised as relevant to the search topic, 8% identified to contain quality information with approximately 0.7% as "high-quality" Web-pages. The high proportion of relevant pages is a result of close semantic distance between the search topic and the chosen queries.

Grounded theory analysis explores how well the information from the extracted Web-pages expands the search topic. Figure 6.3 displays cumulative KU conversion rate across five queries. Analysing the KU conversion rate throughout the search process, KU function does not tend to convert, meaning that the five iterations done were not enough to exhaust the search topic. Thus, there is a need to iterate further applying different queries to extract more information.

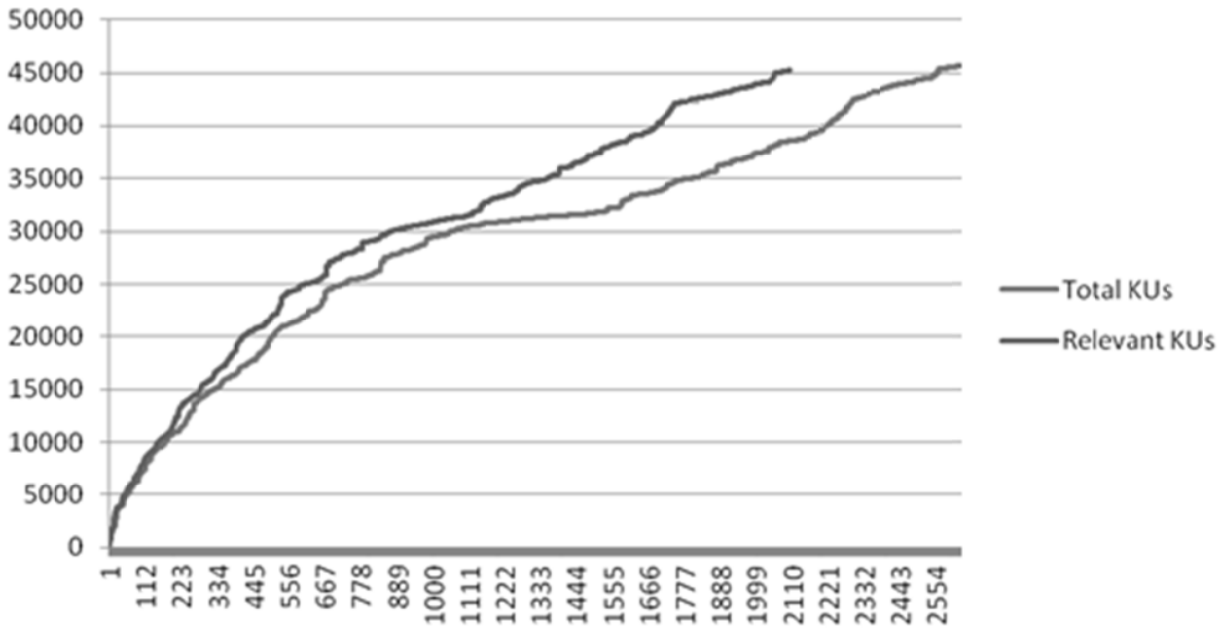


Figure 6.3 – Cumulative KU conversion rate across five queries (Experiment 1)

This experiment resulted in the notable improvement in quality of the pages placed towards the top of the final list of results compared to the Google PageRank order results. Figure 6.4 displays quality change for top search results. The average quality at the top of the Google list was estimated as 5%, while the top of the highest positioned Web-pages suggested by the system had an average quality of 28% with the average quality of highest positioned documents of 78%.

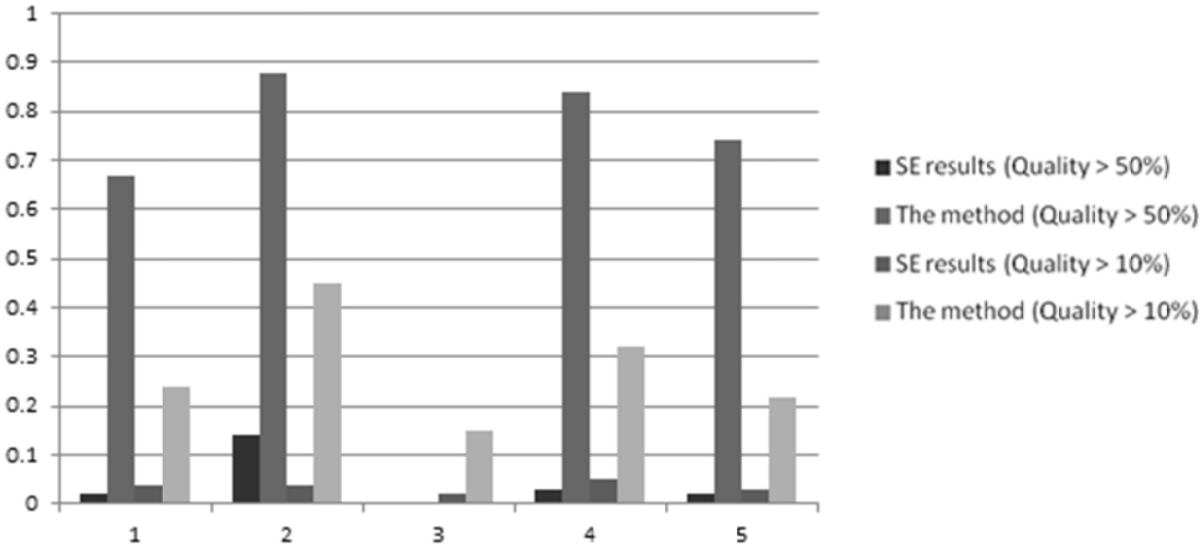


Figure 6.4 – Quality change for top search results (Experiment 1)

The resulting list of Web-pages did not follow the order of the Google PageRank algorithm and had Web-pages from the middle and end of Google list closer to the top of the reordered list of results. This indicates no correlation exists between the Web-page quality and the depth of the search results list.

6.4.2 Experiment 2 – Same well-defined knowledge base

During the second experiment using the search topic "cocaine smuggling" the performance of the system showed similar results. Again, this experiment is aimed to test the performance of the system when each new Web-page content is compared against the same knowledge base initiated at the beginning, and it resulted in the significant reduction of information provided by Google search across all five queries.

To explain further, there were 552 Google search results for Q1 "cocaine production and distribution", 472 of them were considered relevant, 95 Web-pages recognised as quality information with 21 "high-quality" pages. Q2 "cocaine trafficking UK" had 631 results in Google list, 514 of them were relevant at the NLP stage, 30 were recognised as containing quality information with 2 "high-quality" pages. The list for Q3 "cocaine smuggling UK" had 555 Web-pages, 440 were relevant and 11 pages were classified as quality information. Q4 "cocaine cartels" had 739 Web-pages, 598 of them were relevant and 53 were quality information with 6 pages considered as "high-quality". Q5 "Mexican drug cartels" had 426

Web-pages, 353 of them were relevant and were quality information with 5 pages considered as “high-quality”. Figure 6.5 demonstrates distribution of semantic closeness values across five queries for this experiment.

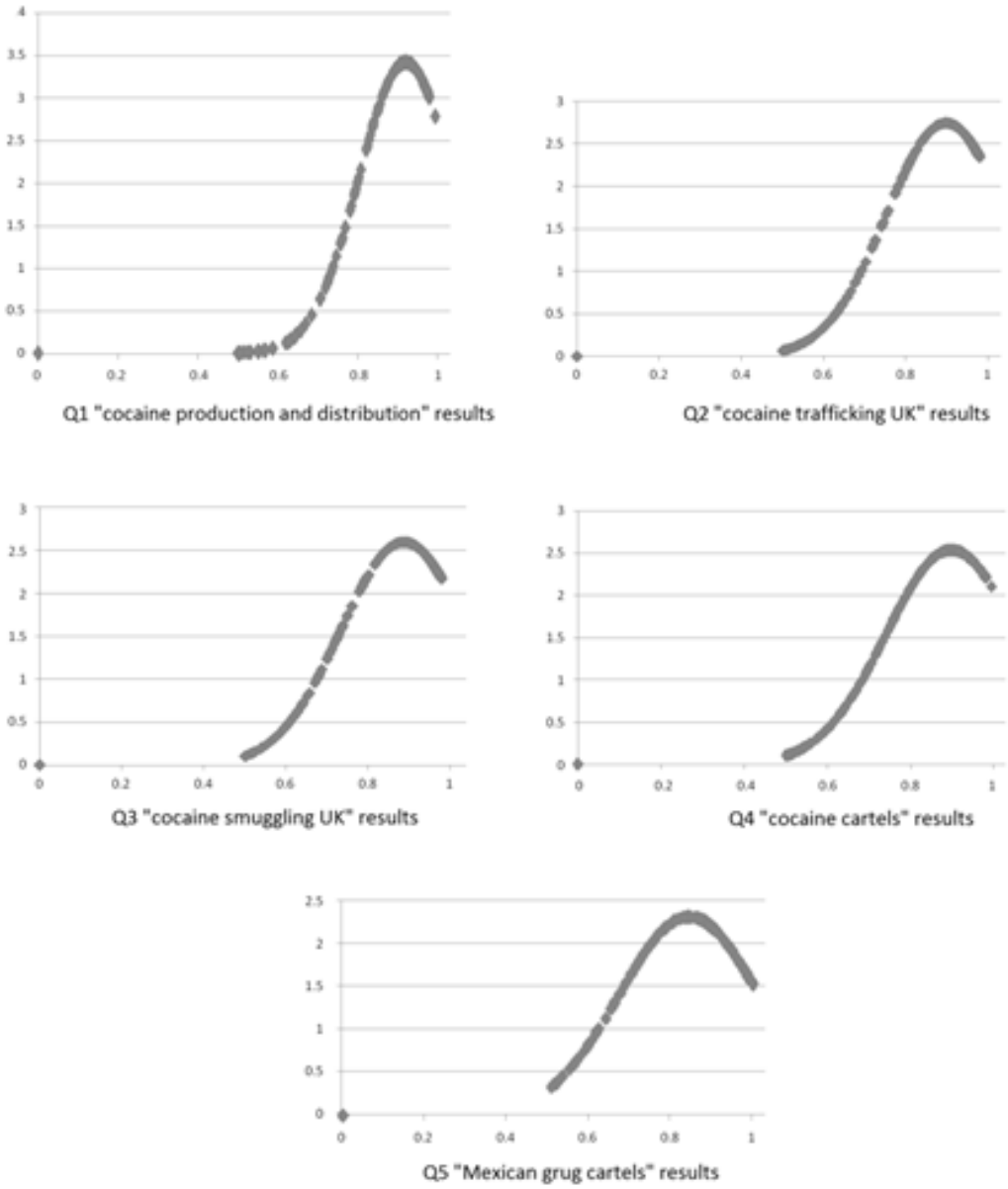


Figure 6.5 – Distribution of semantic closeness values across five queries (Experiment 2)

Thus, out of 2,903 Web-pages suggested by Google across all four queries 82% were recognised as relevant, 7% were identified to contain quality information with 1% as “high-quality” Web-pages. The increased proportion of the relevant pages can be explained as a result of close semantic distance between the search topic and the queries.

Grounded theory analysis provides evidence on how well the information from the extracted Web-pages expands the search topic (Figure 6.6).

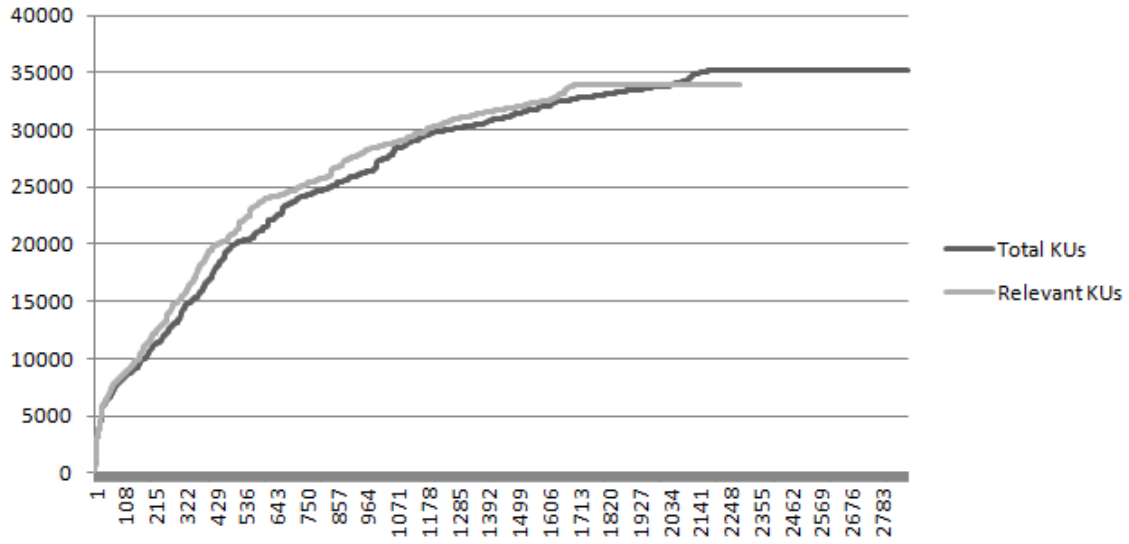


Figure 6.6 – Cumulative KU conversion rate across five queries (Experiment 2)

Analysing the KU conversion rate throughout the five iterations the cumulative KU function becomes close to convergence after about ¾ of the processed Web-pages. This can indicate that the information obtained with the chosen queries spans the search topic well and new Web-pages will not add a significant amount of new information.

Figure 6.7 shows reduction in Web-pages amount after three stages of filtering.

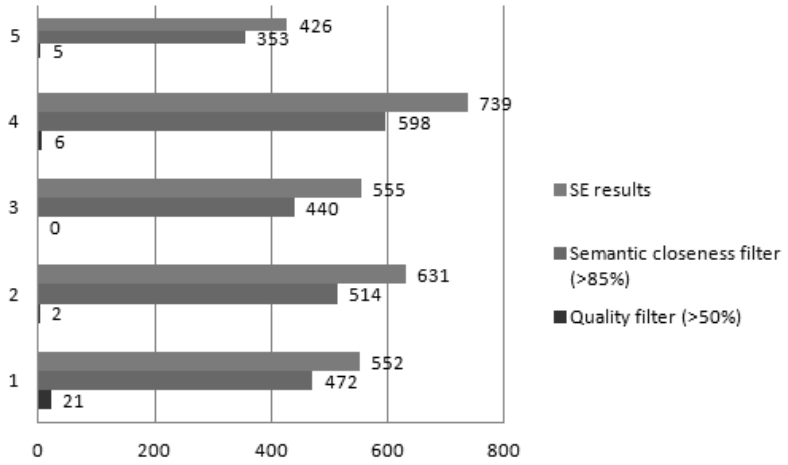


Figure 6.7 – Reduction in Web-pages amount after three stages of filtering (Experiment 2)

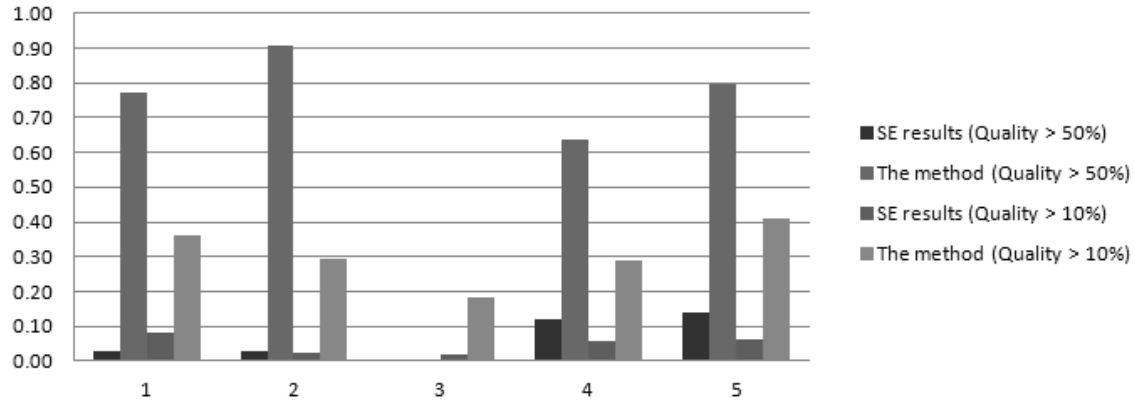


Figure 6.8 – Quality change for top search results (Experiment 2)

The quality of the top rated pages after the five iterations has improved compared to the Google PageRank order results (Figure 6.8).

The average quality at the top of the list increased from 4% to 28%, with the average quality of highest positioned Web-pages of 77%. Noteworthy, the resulting list of Web-pages did not follow the order of Google PageRank and had Web-pages from the middle and end of Google list closer to the top of the reordered list of results. This indicates no correlation exists between the Web-page quality and the depth of the search results list.

6.4.3 Experiment 3 – Appended well-defined knowledge base

This experiment used the search topic “tobacco industry”, but this time the initial knowledge base was appended with extracted text of the semantically close Web-pages. This experiment also resulted in the notable reduction of information provided by Google search across all five queries. Out of 715 results displayed by Google for Q1 “tobacco industry competition”, 560 were selected after the Semantic Closeness stage and 66 results passed the final quality check test, 5 of which were considered as “high quality”. Q2 “tobacco criminal” returned 413 Google results, 343 of which were passed through after the NLP stage and 20 passed the quality test, only 5 were considered as “high quality”. Q3 “tobacco financial market” had 455 Web-pages, 125 were further selected as relevant, 17 were considered as quality information with none of the Web-pages identified as “high-quality”. Q4 “tobacco smoking statistics” had 855 Google search results, 99 were selected after the semantic filtering stage, 7 were considered as quality information with 2 Web-pages containing “high-quality” information. Q5 “tobacco investment opportunities” had 127 links to Web-pages, 93

were selected after the NLP stage, only 2 Web-pages were identified as quality information, none of them were “high-quality”. Figure 6.9 shows distribution of semantic closeness values across five queries.

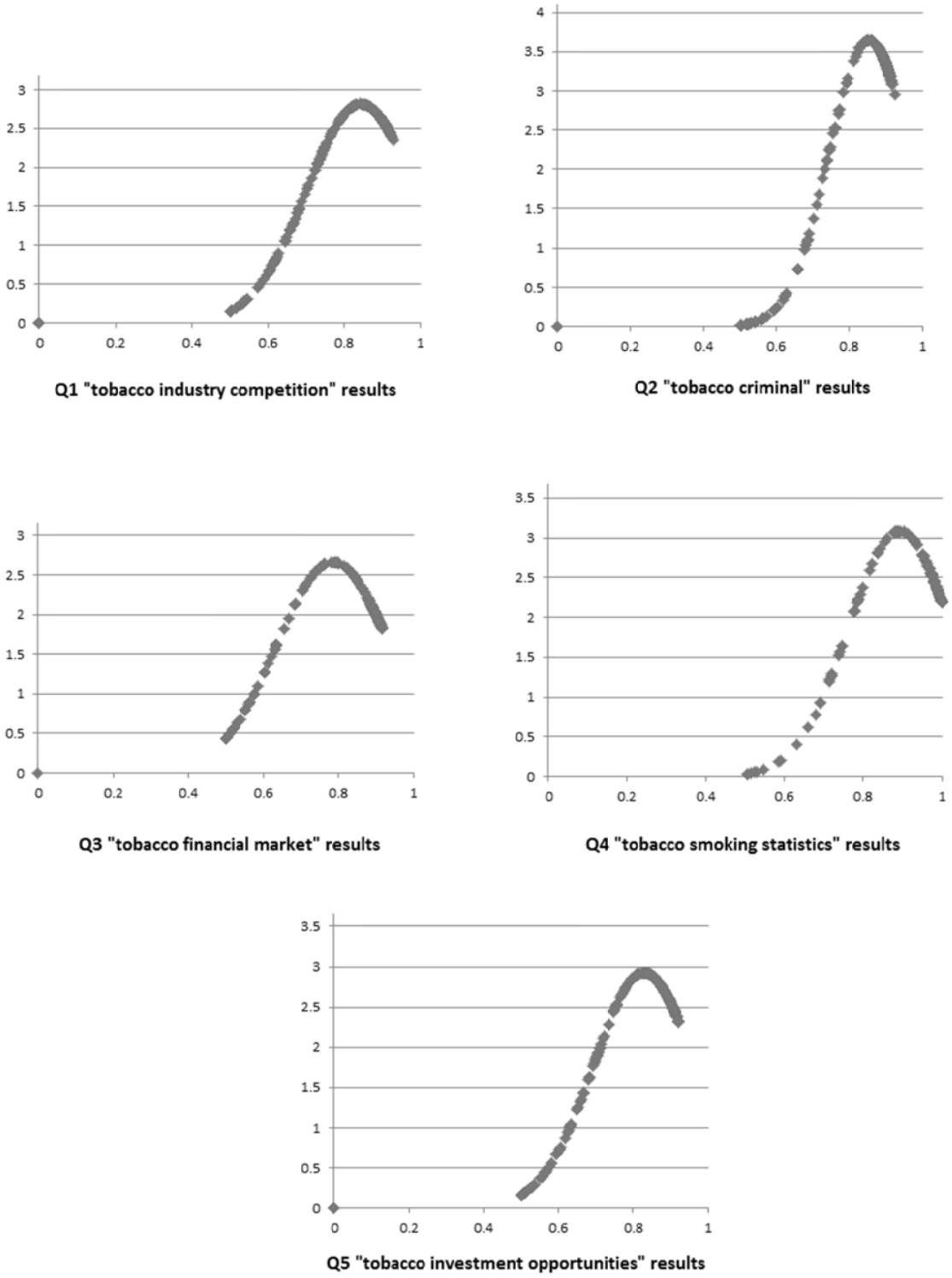


Figure 6.9 – Distribution of semantic closeness values across five queries (Experiment 3)

Thus, out of 2,565 pages suggested by the search engine across all five queries, only 12 pages were considered as relevant information of high quality. This resulted in the overall reduction in information load which was as high as 99.53% of Web-pages, with only 0.47% of relevant high-quality sources. 48% of the Web-pages passed the relevance test, and 7.2% were considered as quality information.

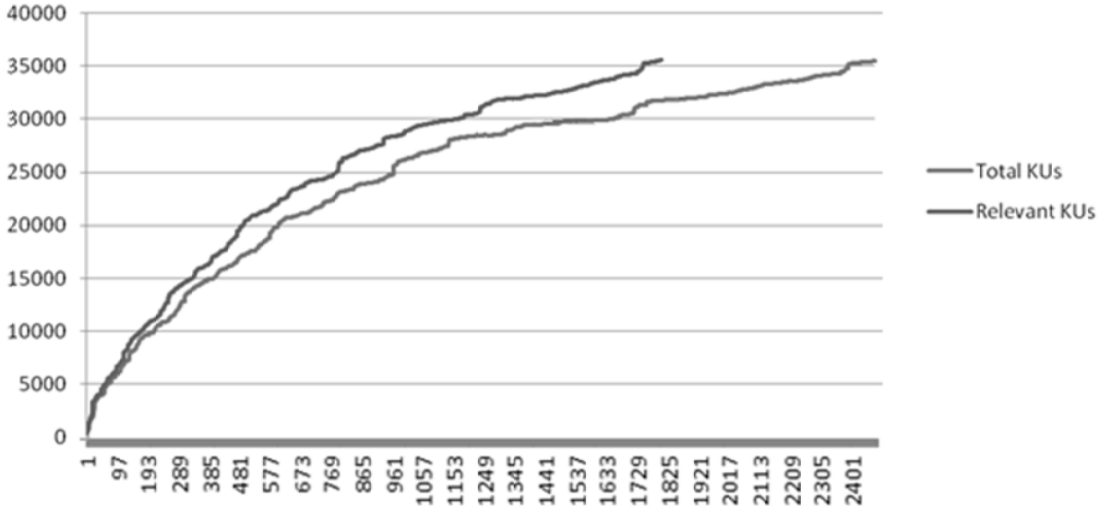


Figure 6.10 – Cumulative KU conversion rate across five queries (Experiment 3)

Grounded theory analysis for the five iterations shows that the cumulative KU function continuously grows throughout the search process (figure 6.10). This indicates the need for further iterating with different queries to expand the search direction and get new information. Figure 6.11 shows reduction in Web-pages amount after three stages of filtering.

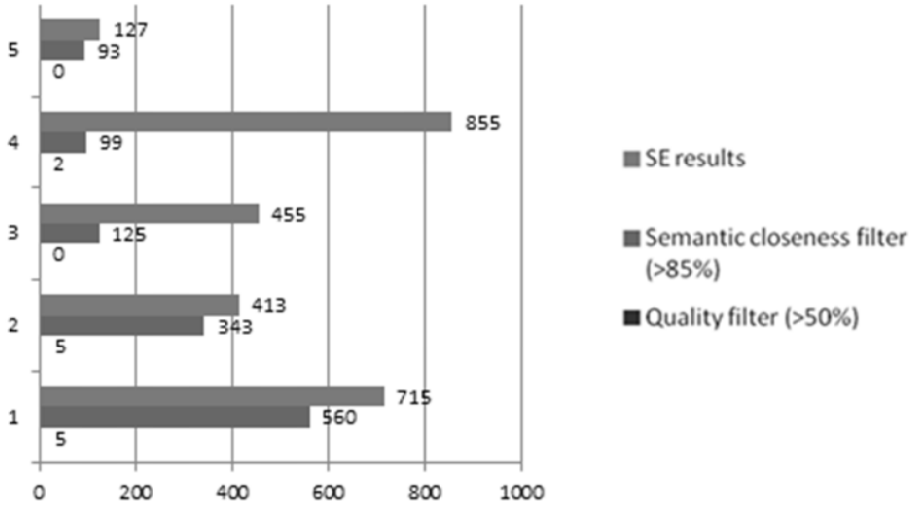


Figure 6.11 – Reduction in Web-pages amount after three stages of filtering (Experiment 3)

The average quality of Web-pages positioned on top of SE results list was estimated as 3.5%. After the five iterations (Figure 6.12), the list of suggested Web-pages resulted in the average quality of 30% at the top of the list with the average highest quality of 80%.

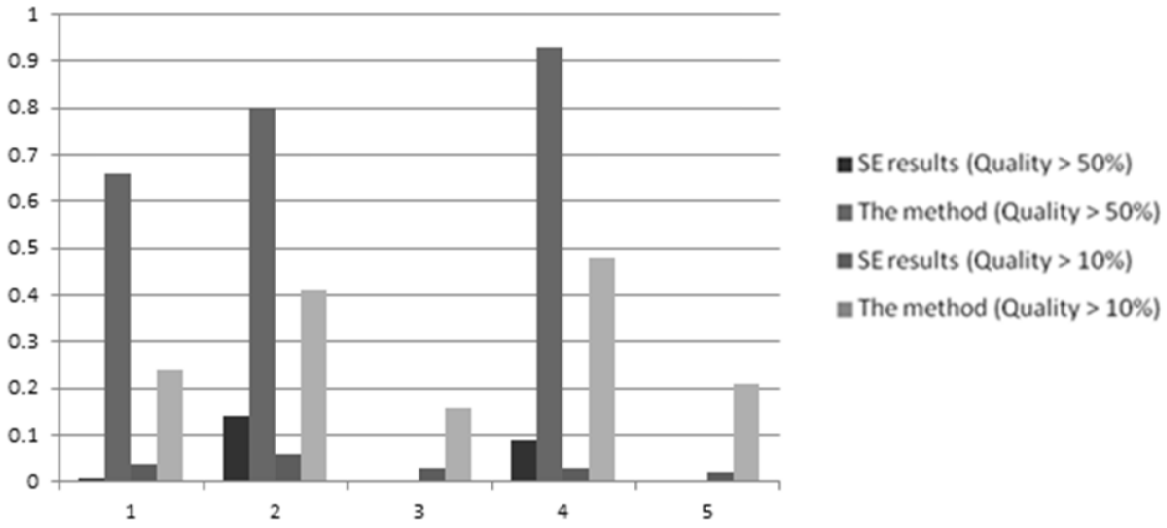


Figure 6.12 – Quality change for top search results (Experiment 3)

Interestingly, but in this experiment the quality of Web-pages content correlated with the depth of the Google search list, i.e. even after reducing the list and changing the order of Web-pages at the final stage, the new order still followed the Google PageRank order.

6.4.4 Experiment 4 – Same poorly-defined knowledge base

This experiment was aimed at assessing how well the system filters relevant Web-pages assuming the level of uncertainty on the subject is high, thus, the chosen queries are semantically remote from the search topic.

This experiment was carried out with regard to the search topic “coffee production”. When the experiment was conducted using the same initial KB, five queries were taken from the “tobacco industry” search topic. The amount of SE results varied for the same queries due to the different time of the experiments and the fact that Google constantly updates its index base. For this part of the experiment Q1 “tobacco industry competition” had 715 results in the SE list, 558 of them were relevant and 62 were considered as quality information with 4

"high-quality" pages. Q2 "tobacco criminal" had 413 Web-pages, 332 pages were relevant, 19 pages passed the quality test and 7 pages were rated as "high-quality". Out of 455 results for Q3 "tobacco stock market" 248 were relevant with 16 pages were above the quality threshold and no Web-pages of high quality. Q4 "tobacco smoking statistics" had 855 results, 651 of them were relevant, 64 contained quality information with 16 Web-pages being "high-quality". Q5 "tobacco investment opportunities" had 127 results in the SE list, 90 were relevant, only 4 were considered as quality information.

The reduction in the amount of pages after the model run is shown in the Figure 6.13 below.

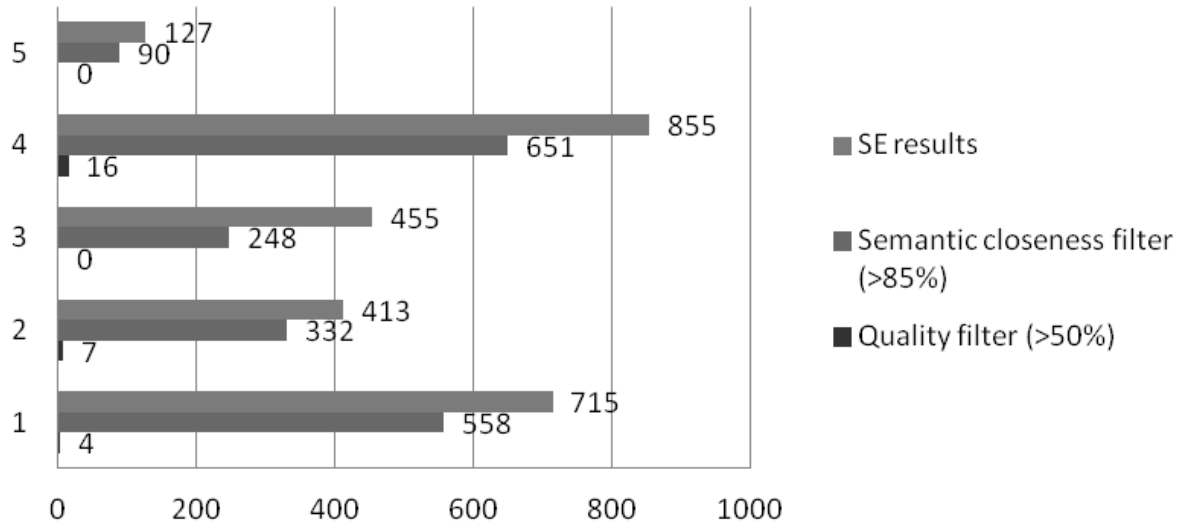


Figure 6.13 – Reduction in Web-pages amount after three stages of filtering (Experiment 4)

The distribution of semantic closeness results across five queries is shown in Figure 6.14 below.

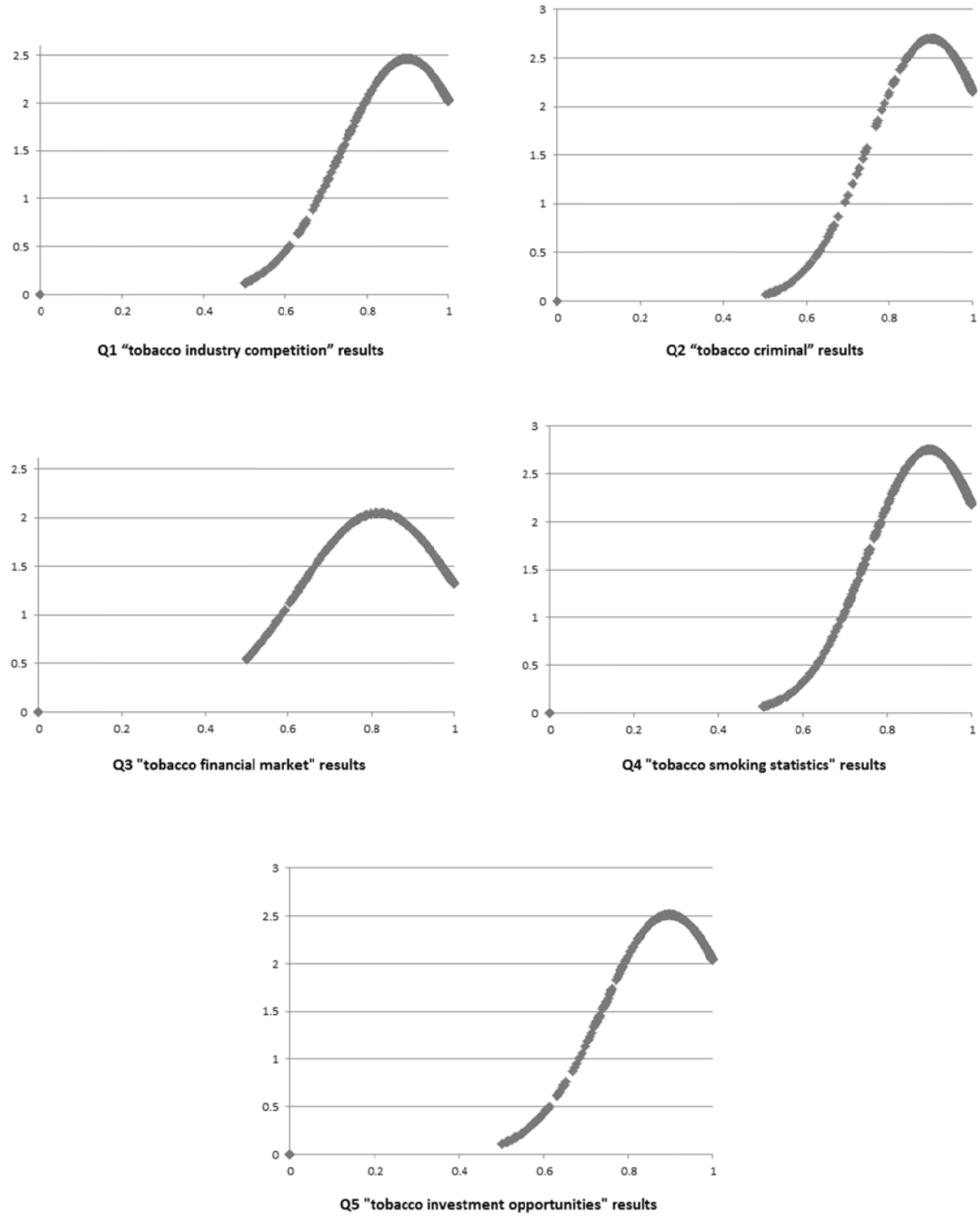


Figure 6.14 – Distribution of semantic closeness values across five queries (Experiment 4)

To summarise, the total number of Web-pages suggested by Google across five queries was 2565, 73% were identified as relevant information and 6% as quality information. Approximately 1.05% of all results were rated as “high-quality” information sources.

Grounded theory analysis for the five iterations shows that the cumulative KU function continuously grows throughout the whole search procedure (Figure 6.15). This indicates that the chosen queries do not cover the search topic well enough and iterating further is suggested.

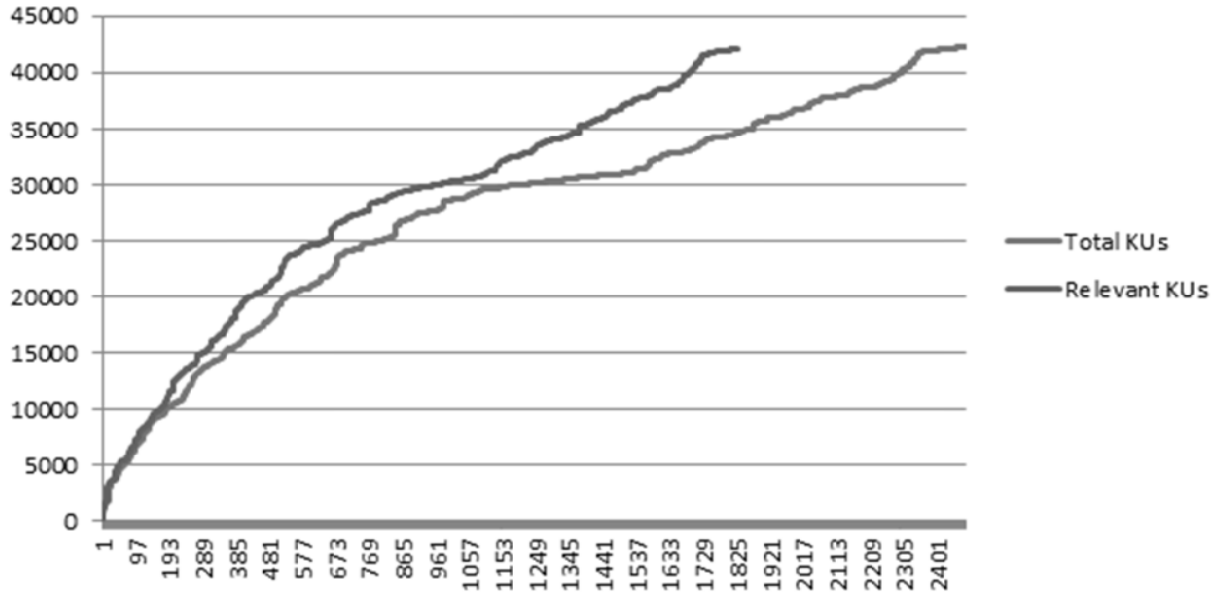


Figure 6.15 – Cumulative KU conversion rate across five queries (Experiment 4)

After the list of Web-pages had been reordered, there was a significant quality improvement in the top of the list. The average quality of top-rated pages from the initial search engine results list was 3.5% increased to 29% in the resulting list with the average quality of the highest rated Web-pages of 75%. The relatively high proportion of the relevant pages for different topics of the knowledge base and the corresponding queries can be explained as a small level of semantic remoteness between the topics, i.e. although “coffee” and “tobacco” have different meanings, these topics have a lot in common.

Figure 6.16 shows the quality change for top search results in this experiment.

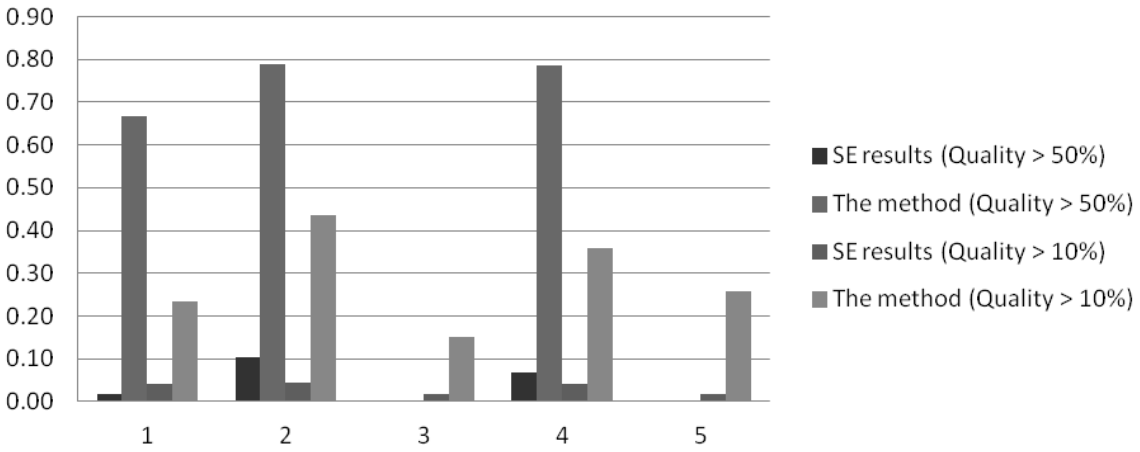


Figure 6.16 – Quality change for top search results (Experiment 4)

The relatively high proportion of the relevant pages can be explained as a small level of semantic remoteness between the topics, i.e. “coffee” and “tobacco” have a lot in common, but have different meanings. The reordered search results list did not follow the PageRank order.

6.4.5 Experiment 5 – Appended poorly-defined knowledge base

The last experiment was evaluated with the idea of poorly defined search topic. For this experiment search queries for the “coffee production” topic were used to find relevant information for the “tobacco production” knowledge base, which was appended with the relevant Web-pages extracted text, meaning that the next page content will be compared against the expanded knowledge base.

The algorithm output was as follows. Q1 “coffee industry” returned 347 results, 292 were considered relevant, 56 of them contained quality information with 20 pages identified as “high-quality”. Q2 “coffee criminal” had 261 results in the search engine list, 177 were relevant after the semantic closeness test, 23 contained quality information with 2 “high-quality” pages. Q3 “coffee financial market” returned 580 search engine results, 347 were identified as relevant, 35 pages passed the quality test with 4 pages identified as containing “high-quality” information.

Distribution of semantic closeness values across three queries is shown in Figure 6.17.

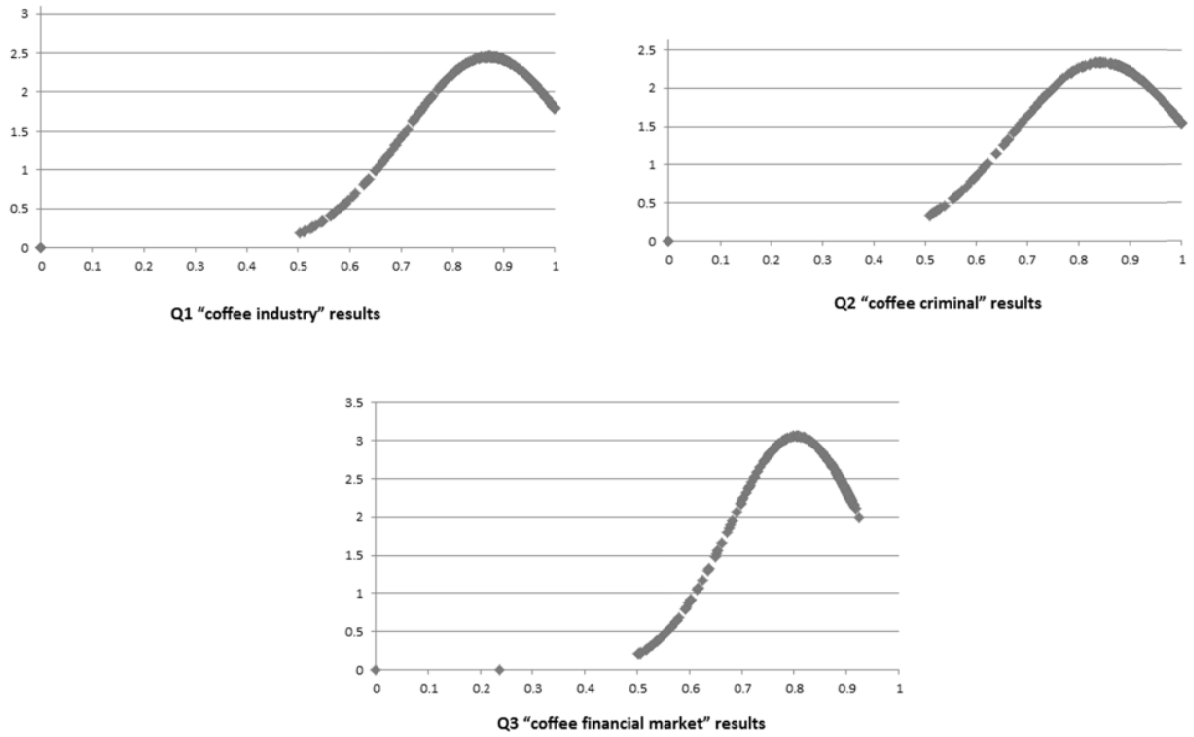


Figure 6.17 – Distribution of semantic closeness values across three queries (Experiment 5)

To summarise, out of total of 1188 Google results for the three queries 68% were identified as containing information relevant to the search topic, 10% of the pages contained quality information with 2% of "high-quality" pages. As with the previous experiments, the filtered list of results did not follow the PageRank order of Web-pages.

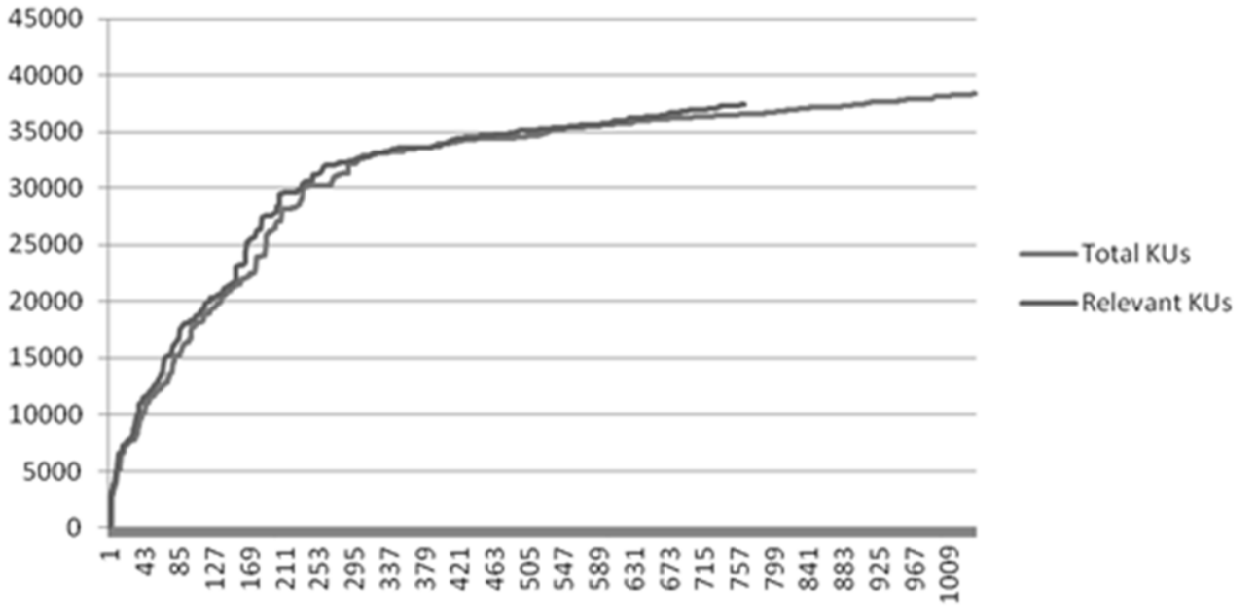


Figure 6.18 – Cumulative KU conversion rate across five queries (Experiment 5)

Analysing the KU conversion rate for the three iterations there is a notable reduction in growth rate of the cumulative KU function after approximately 1/3 of the processed Web-pages (Figure 6.18). This growth reduction may indicate that the search topic is close to exhaustion and that further iterations depending on the chosen queries may or may not add information to the search topic. Figure 6.19 shows reduction in Web-pages amount after three stages of filtering.

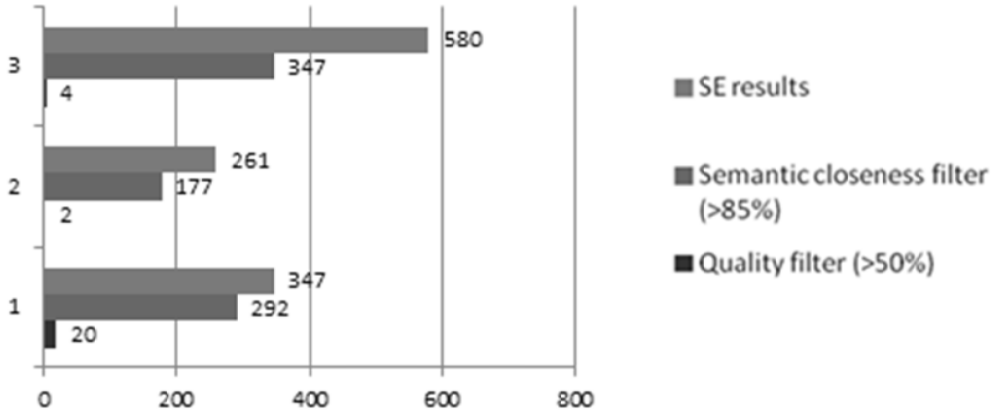


Figure 6.19 – Reduction in Web-pages amount after three stages of filtering (Experiment 5)

The quality level of the Web-pages at the top of the list was significantly improved (Figure 6.20) from the average of 4% among the search engine results list to 31% in the final list of results with the average quality of 78% for the highest rated Web-pages.

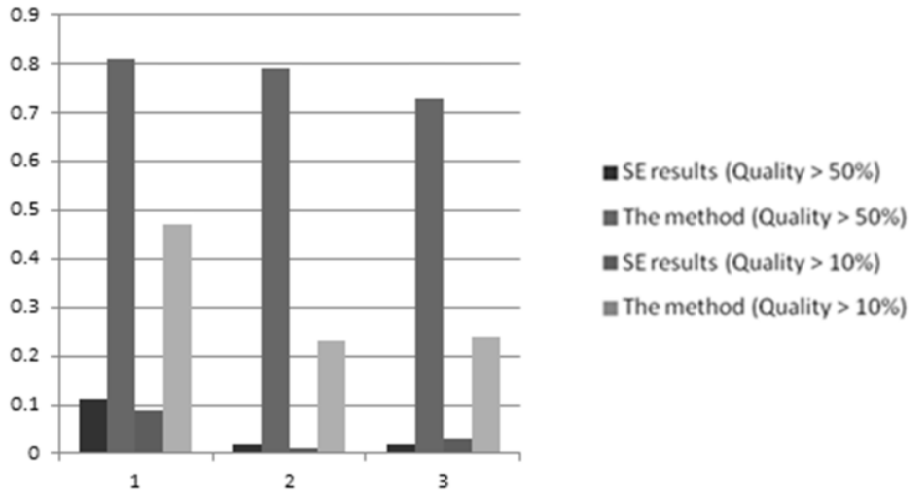


Figure 6.20 – Quality change for top search results (Experiment 5)

Similarly to the previous experiment, the relatively high proportion of the relevant pages can be explained as a small level of semantic remoteness between the topics, i.e. “coffee” and “tobacco” have a lot in common. Again, the newly processed results list did not follow the PageRank order.

6.5 Final results

The results of the experiments are summarised in the table 6.3 below.

Search Engine Results, #		$sim(T_1,T_2)$, # (> 0.85)	Quality Test Results, #		Top Results Average Quality, %			
					Search Engine		System	
			> 0.50	> 0.10	> 0.50	> 0.10	> 0.50	> 0.10
Experiment 1 – Same well-defined knowledge base "tobacco industry"								
tobacco industry competition	715	572	3	68	0.02	0.04	0.67	0.24
tobacco criminal	413	345	5	19	0.14	0.04	0.88	0.45
tobacco financial market	455	248	0	18	-	0.02	-	0.15
tobacco smoking statistics	855	650	8	57	0.03	0.05	0.84	0.32
tobacco investment opportunities	127	101	1	31	0.02	0.03	0.74	0.22
		75%	**0.66%**	8%				
Experiment 2 – Same well-defined knowledge base "cocaine smuggling"								
cocaine production and distribution	552	472	21	95	0.03	0.08	0.77	0.36
cocaine trafficking UK	631	514	2	30	0.03	0.02	0.91	0.29
cocaine smuggling UK	555	440	0	11	-	0.02	-	0.18
cocaine cartels	739	598	6	53	0.12	0.06	0.64	0.29
Mexican drug cartels	426	353	5	19	0.14	0.06	0.8	0.41
		82%	**1.17%**	7.2%				
Experiment 3 – Appended well-defined knowledge base "tobacco industry"								
tobacco industry competition	715	560	5	66	0.01	0.04	0.66	0.24
tobacco criminal	413	343	5	20	0.14	0.06	0.8	0.41
tobacco financial market	455	125	0	17	-	0.03	-	0.16
tobacco smoking statistics	855	99	2	7	0.09	0.03	0.93	0.48
tobacco investment opportunities	127	93	0	2	-	0.02	-	0.21
		48%	**0.47%**	4%				
Experiment 4 – Same poorly-defined knowledge base "coffee production"								
tobacco industry competition	715	558	4	62	0.02	0.04	0.67	0.23
tobacco criminal	413	332	7	19	0.1	0.04	0.79	0.44
tobacco financial market	455	248	0	16	-	0.02	-	0.15
tobacco smoking statistics	855	651	16	64	0.07	0.04	0.78	0.36
tobacco investment opportunities	127	90	0	4	-	0.02	-	0.26
		73%	**1.05%**	6%				
Experiment 5 – Appended poorly-defined knowledge base "tobacco industry"								
coffee industry	347	292	20	56	0.11	0.09	0.81	0.47
coffee criminal	261	177	2	23	0.02	0.01	0.79	0.23
coffee financial market	580	347	4	35	0.02	0.03	0.73	0.24
		69%	**2.19%**	10%				

Table 6.3 – Summary of the experiment results

6.6 Summary

This chapter provided modelling results from testing together with results from five experiments that have been conducted for the purpose of this research.

During the model testing and exercising the research-developed algorithm, the system generated results were compared with human manual analysis results. A random sample of Google search results was manually reviewed and assigned with either Relevant or Not Relevant tag. The testing results showed that system can recognise relevant content with precision 67% and recall 93%.

Five experiments were run using three search topics. The experiments had an objective of assessing the system's performance on finding topic-related information of high quality from the Web.

Two cases have been considered. First part of the experiments used search topics which were well defined with the corresponding queries focused around the initial knowledge base.

The second part of the experiments considered testing the system's performance when the topic understanding was incomplete and, thus, the chosen queries were remote from the search topic defined in the initial knowledge base text.

For all three search topics the conversion rate of KUs was taken as a parameter for identifying when the iterating should stop. The cumulative sum of knew words was analysed and, if little new information was added to the KB, it was assumed that the topic was close to be exhausted and new iterations on the same topic would not significantly enhance the KB from the semantic viewpoint.

Results of the experiments were summarised in a table and illustrated with various graphs. The graphs included distribution of semantic closeness values across all queries, cumulative KU conversion rate across all queries, reduction in Web-pages amount after all stages of filtering, and quality change for search results placed at the top of the list.

Chapter 7 discusses the results of the above experiments and the overall findings of this research.

Chapter 7

Discussion

7.1 Overview to Chapter

This chapter explores how the research objective was addressed throughout the developed research method. The aim of this chapter is to show that the quality of collected information can be significantly enhanced with the use of the research-developed algorithm.

Section 7.2 will outline key research assumptions and findings in accordance with the set research method, Section 7.3 will summarise the key results.

7.2 Research Overview and Implications

Recall the research objective:

To develop an algorithm that supports topic related search for Web-based information and estimates the quality of the extracted content, proving it is significantly enhanced comparing to the traditional Web-search approaches.

In order to meet the research objective, a research method was developed. The following sections will address the research method steps, and discuss the implications of the research and application of the results presented in the previous chapters.

7.2.1. Research Step 1

Investigate the structure of the Web

This research sees the Web as the richest source of digitally stored information. Selection of current, relevant and trustworthy data for a decision information gathering task (English, 2009) can be complicated, since it is getting harder to find relevant information of high value and are no quality control procedures for information uploaded. Web data remains

duplicated and highly contradictory. As section 2.2.1 discussed, a rapid expansion of available Web data implies an important issue associated with evaluating trust in the published sources.

There Web can be split into three layers: Surface Web, Deep Web and Dark Web as shown in figure 2.7. Traditional search engines can access the Surface Web and a small part of the Deep Web. Dark Web cannot be accessed through conventional means.

With all the richness of the Web as an information source, the overall structure of the Web does not allow enough freedom for obtaining knowledge. The information on the Web-sites is stored in silos. Search engine optimisation (SEO) of the Web-pages content puts further focus on the keywords and treats the Web content as bag of words, rather than ideas with context, making discovery of unknowns an issue.

7.2.2. Research Step 2

> *Investigate the information gathering process and the ways of information representation in order to ascertain why richness cannot be accessed via a direct keyword based search. Analyse how information about the real world is stored and accessed, and what effect the limited access to that information may have on decision making.*

The rich picture 2.6 presented a view on the information flow related to the process of decision making. Section 2.3 assumed the parallel existence of two major information sources – the Real World and the Global Knowledge Base. If a decision has to be made about a situation in the dynamically changing Real World, the only information source available to analysts is the information stored in the persistent Global Knowledge Base. Limited access to information similarly to the information overload will result in lower quality of decision making, as shown in figure 2.2.

According to Ashby's Law of Requisite Variety (discussed in section 2.2.2), tools that are used for information gathering need to provide enough amplification for the variety of the extracted information to be as close to the variety of the Real World situation as possible for the most accurate decision. However, traditional keyword search approach does not comply with this law. The search on a complex query often fails – it is quite difficult, even impossible, to predict how information about a situation was transformed by the

publisher before it got stored on the Web. Thus, there is a need for a different approach that overcomes this limitation associated with keyword matching technique.

7.2.3. Research Step 3

> *Explore the ways of representing and estimating uncertainty on a search topic, as well as how to track the reduction of uncertainty during information discovery.*

Rumsfeld's philosophy was introduced in Section 2.2.3. Based on his concepts of known knowns (KKs), known unknowns (KUs) and unknown unknowns (UUs), it was assumed that any topic on any subject can be split into KK, KU and UU parts with the constant total knowledge size and changing proportion of the KK, KU and UU parts. The amount of uncertainty is unique for every individual and resides in the KK and KU parts. Uncertainty depends on the depths of an individual's knowledge on the subject.

This research assumes that the proportion of uncertainty on a topic changes with information discovery due to KUs and UUs being converted into KKs. Such conversion of "unknowns" into "knowns" is the main purpose of information discovery and is one of the major ideas that lie behind this research. The conversion rate is seen as a method of tracking the reduction of uncertainty during information discovery, addressing the issue of keeping the level of information within the boundaries for effective decision support.

7.2.4. Research Step 4

> *Investigate how current search engines operate and to establish why they cannot fulfil a requirement to identify and retrieve information. Explore the methods used for the Web search as well as enterprise search, and identify their benefits and limitations, especially when targeted at the discovery of unknowns.*

This research explored the techniques that are widely used for searching the Web content (section 2.4.2) as well as enterprise search solutions (section 2.4.3).

Latent Semantic Indexing (on example of Google search engine) is seen as an effective Web solution that provides fast results on short queries. As it involves matching two sets of keywords, not only irrelevant information gets responsive to the search query, but also part of the relative information is often overlooked, if explained with different words.

However Google constantly improve their search mechanism by introducing elements of the semantic search, such as synonyms. PageRank algorithm is seen as a quality measure that is based on the popularity of a Web-page. Keyword matching search technique paired up with the query-independent PageRank algorithm results in the chaotic mixture of relevant and irrelevant Web-content, ignoring the important relevance aspect.

Ontologies solve this relevance issue as they provide shared representations of the entities and relationships characterising a search topic domain. These shared vocabularies can be used to observe and record Web users' behaviour to help in articulating what they want to search for in the future. The scope of the web does not allow ontologies to be applicable for a broad Web search as they are usually limited to a selected set of search topics or information resources. In the enterprise search world ontologies are widely used to support query answering against distributed and/or heterogeneous data sources. The main limitation is that ontologies and typical queries are often fixed at application design time, raising a concern over the completeness of the extracted information, or in other words how well the information spans the search topic.

Meaning-based computing (on example of Autonomy solution) is seen as an effective statistical approach to enterprise search. The scope of an enterprise knowledge base allows sophisticated mathematical algorithms to sort and tag the content within it and return results quickly. Such complex approach was concluded to be not feasible when applied to the Web due to high computational complexity.

This research also explored Theory of strongly semantic information (Floridi, 2004) – solution that is aimed at addressing the completeness issue of information and focused on calculating semantic content based on the truth values as opposed to the traditional probabilistic approach. The approach does not accept contradictory or false content as informative and allows more accurate qualification of the content. However, qualification of the content as truthful means identification and qualification of all possible states that correspond to the given situation. This makes Floridi's approach too theoretical to be applicable to the real world situations.

The above methods allow estimating relevance and completeness of the search results. Only limited research has been carried out with regards to measuring the quality of the gathered information and all of such methods (reviewed in chapter 2) have limitations in terms of information sources or functionality.

Thus, after exploring the problem and existing solutions further, it was concluded that available approaches lack one or more of the fundamental metrics for Web-based

information extraction: relevance, completeness and quality. Modification of existing theories can enable the information content to be tested and allow harvesting Web-wide data in accordance with these three controlled parameters in order to subsequently transfer this data to a knowledge base where enterprise search technologies may be applied in the traditional way to enhance decision making support.

7.2.5. Research Step 5

Investigate research concepts and methods that can be used (or further developed to be used) with a conventional search engine to enrich information retrieval through the discovery of unknown unknowns.

1) Relevance Aspect

Due to the discovered need to systematically identify the semantic relationship between the search topic and the Web-content, the research step 7 looks at the methods that would provide "*an effective approach to formulate a topic-focused collection of the Web content that is semantically related to the search topic"*.

Part of this research has investigated the ability of a traditional search engine to find relevant information based on the keywords match technique (section 2.4.2). In order to address the issue of semantic relevance, the proposed search model relies on the Google search engine as it able to access a large part of the Web. The search results are evaluated against a search topic that is defined in form of unstructured text.

A combination of existing approaches that use both, statistical and semantic methods, to analysing texts for semantic relatedness was chosen to address the relevance aspect of the new search algorithm. First, the method by Hirst & Mohammad (2006) is applied to calculate concept-to-concept similarity. This measure is then applied to the technique proposed by Corley & Mihalcea (2005), replacing the word-to-word similarity in their original formula with concept-to-concept similarity values in order to calculate the semantic similarity of two texts. Such technique allows filtering of Google search results by removing false positives.

2) Completeness Aspect

Next, addressing the completeness aspect of the Web-based information extraction, the step 7) of the research method looked at the techniques that would provide "*a mechanism*

for tracking the growth of information and the reduction of uncertainty during information discovery".

Grounded theory has been chosen to test the completeness of gathered information. Relying on the idea that in the beginning only the information within the predefined initial knowledge base is considered as known ensures the collected knowledge is objective and can be explicitly defined, while grounded theory provides a systematic approach to building evidence on the chosen search topic. Web pages that passed the semantic closeness filter will mainly consist of new concepts compared to the initial knowledge base. These new concepts are considered as converted to KU and KK. Such conversion rate provides a mechanism for tracking the growth of information together with reduction of the associated uncertainty.

3) Quality Aspect

Finally, addressing the quality aspect for the Web-based information extraction, the step 7) of the research method required investigation of the techniques that would provide "*a quality measurement of the extracted content"*.

When the traditional search engine presents its search results, they are ordered in accordance with some Web-page quality criteria. The PageRank algorithm by Google is not query driven and, thus, orders search results based on the Web-page popularity metric, which often does not align with the semantic value of the content.

To overcome this issue, this research suggests using Dempster-Shafer theory for estimating quality of Web-page content from semantic viewpoint, hence, setting a quality parameter for the efficacy of the eventual decision-making. The total quality score for each Web-page is calculated as a Belief function, based on which the final list of results is reordered bringing web-pages of higher quality to the top of the list.

7.2.6. Research Step 6

> *Develop a search model that will integrate a conventional search engine with concepts and methods (the algorithm) that will undertake the search. This would need to be a working model that can be tested using the Web.*

Chapter 3 proposed a new search model (figure 5.3) that combines the benefits of statistical and semantic approaches to information search in order to return results that are semantically close to the selected search topic. It follows with completeness and quality test

to ensure the gathered relevant information is within the boundaries for the high quality decision making, as shown in Figure 3.8. Thus, the proposed solution overcomes the limitations associated with methods described in Chapter 2 and addresses all three aspects of the effective search model. The mathematics for the proposed solution was developed in Chapter 4.

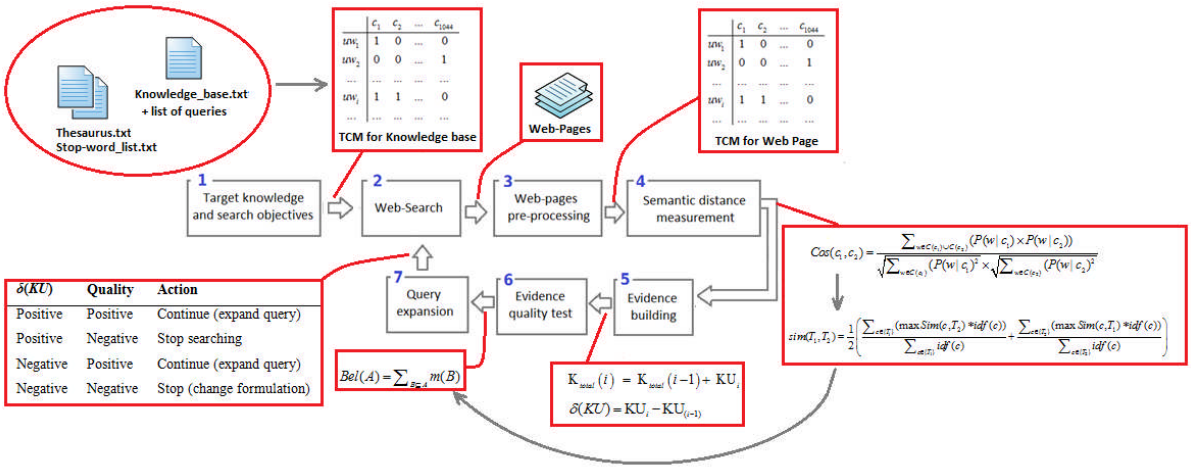


Figure 5.3 – A detailed schematic of the working model

The overall process is iterative and applied to the Web, using Google search engine as an extraction mechanism. A single iteration corresponds to a single search query. Application of Dempster-Shafer theory for analysis of the results within provides a quality parameter that, in contrast to the PageRank algorithm, orders the filtered (relevant) Web-pages in accordance with their semantic value and strength of support by the other relevant Web-pages. This approach brings the potentially overlooked Web-pages closer to top of the search results list, enhancing the discovery of unknown information. The discovered unknowns are used as a basis for new query expansion in order to identify the best information for the target decision process.

7.2.7. Research Step 7

Test the model through use cases and undertake analysis of the results. Develop experiments that would address the efficiency of the model to find the Web content that is relevant to the defined search topic, as well as to track the discovery rate of

unknown unknowns, and to provide a reliable quality metric that would assess the discovered information from the richness viewpoint.

This section will also discuss the results of the five conducted experiments (see Chapter 6) also addressing the three fundamental aspects of Web-based information gathering: relevance, completeness and quality.

1) Relevance aspect

From the relevance perspective, clearly, the better knowledge the analyst has on the search topic, the greater chance there is that the selected keywords formulate the query that will more accurately describe the topic. Moreover, one may apply "advanced search" option and use Boolean operators to improve search results. For the purpose of this research the "advanced search" option was not used.

The use of the Google keyword search was expected to result in search list that contained many false positives. Nevertheless, the experiments showed (see table 6.3) that the major part of the search results (approximately 70%) was still considered relevant to the search topic with relatively high semantic closeness value of 0.85-1.0. The experiments revealed that approximately 30% of the Web-pages in the results list have little or no relevance to the search topic, proving that keyword search brings back nearly a third of Web-pages that contain terms from the query but discuss a completely different subject.

Thus, up to almost one third of the search results may turn out to be irrelevant to the search topic and the experiments have shown that some of these irrelevant Web-pages are placed at the top of the results list.

Figure 7.1 shows an example how the top of the search results list got changed after applying the proposed quality metric (on example of the Experiment 2 Q1 – see Chapter 6, Section 6.4.2).

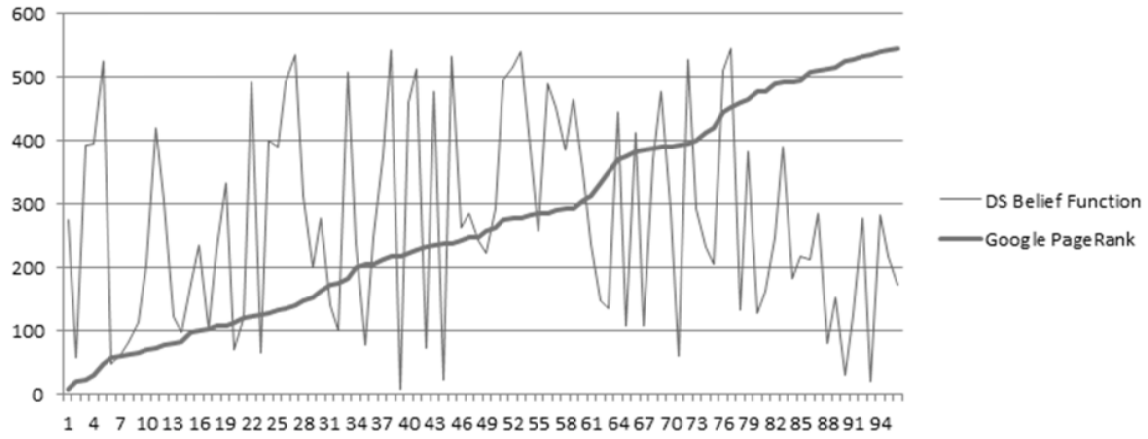


Figure 7.1 – Example of top quality search results reshuffle for Experiment 2 (Q1)

Out of 95 results with the quality score over 10%, the relevant quality information was found as deep as the 546th result in the Google search list. Among the selected quality Web-pages, the highest PageRank scored Web-page (placed 7th in the Google list) moved to position 39, while the lowest PageRank scored Web-page (placed 546th in the Google list) moved to position 77in the list of Web-pages suggested to be transferred to the knowledge base.

Studies show that when analysing the "search engine" results and deciding which Web-pages are relevant, only a rare user goes beyond the first 100 snippets presented by the search engine (Hochstotter & Lewandowski, 2009), (Wai-Ting Leung, et al., 2008). In the above example shown in Figure 7.1 only 16 results were originally positioned in the top 100 of Google search results list. This means that using the traditional keyword search the relevant high quality Web-pages placed towards the end of the long search results list are likely to remain ignored. The proposed solution used with a traditional search engine significantly enhances information retrieval through the discovery of unknown unknowns placed across the long list of search results.

In addition, the analysis of values of the probability density function in the relevance test results may help to understand how well the corresponding query reflects the search topic in the eyes of the search engine. Consider the results of the Experiment 1 (see Chapter 6, Section 6.4.2). Figure 7.2 provides visualises comparison analysis of the semantic closeness test results for Q1 and Q4.

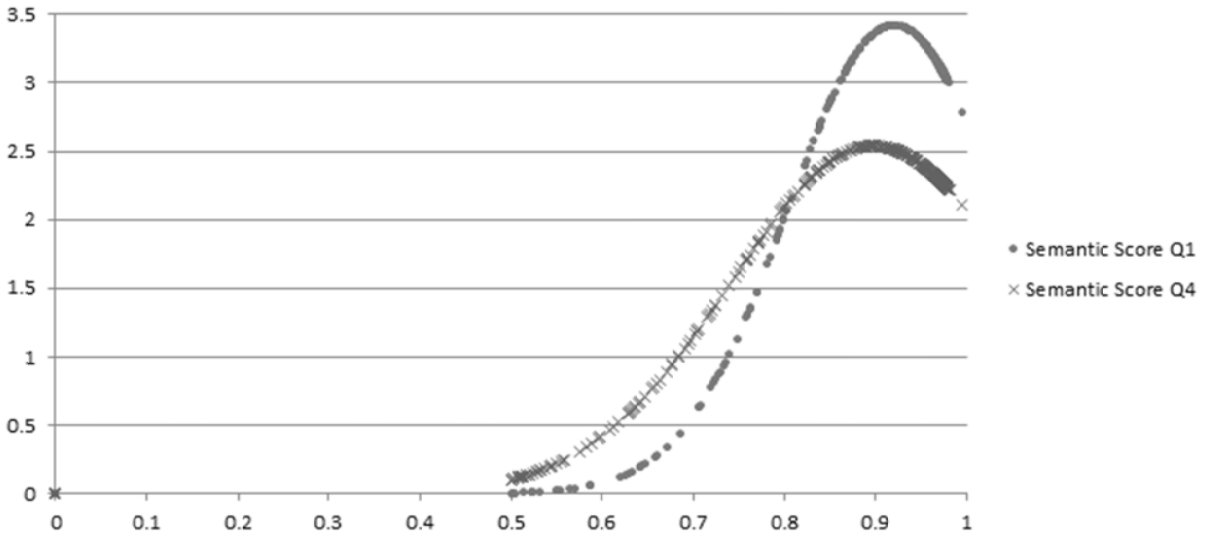


Figure 7.2 – Semantic closeness scores for Experiment 2 results of Q1 and Q4

The narrower bell curve corresponding to Q1 indicates that the chosen query matches the search topic more accurately than Q4. Query Q1 results in more relevant pages after the Web-search. A wider bell curve (query Q4) indicates that the chosen query brings back a broader spectrum of information and expands the search topic.

Therefore, the proposed search method makes it possible to adjust the scope or direction of the Web-search by analysing a visual representation of the semantic closeness score. Precise queries that are themselves highly semantically close to the search topic will fill up the gaps in the knowledge on the subject without expanding it. On the contrast, queries that go beyond the meaning of the search topic will unveil possible directions of the topic expansion. This is the case where the initial knowledge on the topic is limited and, as a result, the queries formed on that limited information can be very remote semantically from the initial search topic.

2) Completeness Aspect

If an analyst views only snippets in the search results list before choosing which Web-pages to open and further explore, the testing part of the experiments has shown that it should take approximately five minutes to review one results page of 50 snippets. Thus, for average 2,500 results across five queries it would take the user approximately 250 minutes or 4 hours and 10 minutes just to look through the snippets without opening the Web-pages.

As for the content on each of the Web-pages, a number of studies have been conducted that explore reading habits of Web users when scanning the Web content (Chaparro et al., 2004), (Nielsen Norman Group, 2006), (Nielsen Norman Group, 2008). These eye-tracking studies on the reading habits of Web users revealed that when skimming a Web-page most users tend to concentrate on the content which is placed towards to the left as well as content that fits an F-shaped pattern, such as headings followed by paragraphs.

Nielsen Norman Group (2008) revealed that people only read 28% of the text on a Web-page and this number decreased the more text there is on the page, meaning that if an important piece of information is placed towards the bottom of a long document, there is a high chance that this information will not be noticed by the user.

Obviously, users tend to spend more time on pages with more information. Based on a linear correlation between the Web-page word count and average time users spend on this page, Nielsen Norman Group estimated the hypothetical maximum number of words users would be able to read, if they spent their entire time on reading while visiting a Web-page. The following chart (Figure 7.3) by Nielsen Norman Group (2008) shows the maximum amount of text users could read during an average visit to Web pages with different word counts.

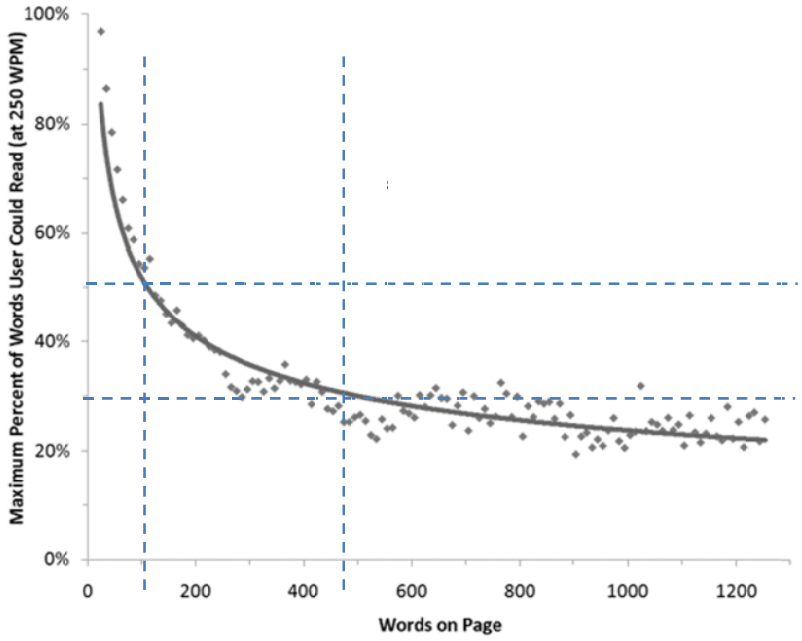


Figure 7.3 – Nielsen Norman Group: A maximum amount of text users read on average visit

From the chart, only half the information is usually read only on those pages with just over 100 words, and for the average Web-page containing 474 words (Levering & Cutler,

2006) the users will read about 30% of the text on it. According to (Chaparro et al., 2004), if the average reading speed is 200 words per minute, it would take 42 seconds for an individual to read 30% of an average 474 word Web page, or to skim three average Web-pages in just over two minutes.

In contrast to the above, the proposed solution in this research overcomes the human factor. Keeping track of the discovered unknowns and their conversion rate by means of Grounded Theory provides a control mechanism for the gathered information expansion. The search method is also developed to analyse the full amount of textual content despite its length, layout or reading age. Taking the example of average 2,500 Web-pages across five queries it would require 1,777 minutes or almost 30 hours to skim only one third of the total content. Considering the unlikely scenario when the user looks through the entirety of the textual content for the example (i.e. 2,500 Web-pages x 474 words = 1,185,000words), it would take 5,925 minutes or nearly 100 hours to look through all the Web-pages for the above example of five queries. While the proposed solution is able to process the same amount of information in approximately 5 hours, which is 20 times faster comparing to the human processing. If the system's code gets optimised, the processing time can be significantly reduced.

3) Quality Aspect

This research refutes the idea that the more popular a Web-page is the more relevant content it contains. In case of the PageRank algorithm, the quality reduction does not always correlate with the depth of search. In only one experiment (Experiment 2) the quality of pages content was decreasing along with the position in the list of search results. This may be the result of a very close semantic distance of the query and the initial knowledge base. In all other experiments there was no correlation between the quality value and the depth of search engine results list.

In addition to the Figure 7.1, the graph below (Figure 7.4) shows an example of how the top quality search results were reordered after the quality stage of the algorithm for Experiment 1 Query 1 (see Chapter 6, Section 6.4.1).

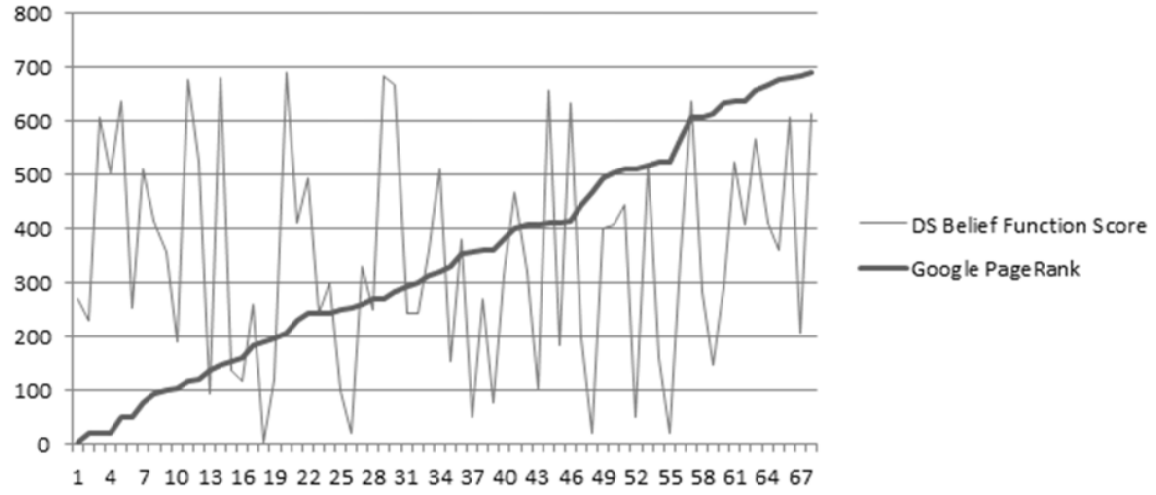


Figure 7.4 – Example of top quality search results reshuffle for Experiment 1 (Q1)

In contrast to the query independent PageRank sorting algorithm, the proposed quality measurement based on the Dempster-Shafer theory employs semantic closeness parameter to estimate the quality of Web-pages for a particular query based on the semantic closeness of a Web page and a formulated search topic, rather than the Web-pages popularity. This technique allows rearranging the search results list, moving the relevant Web-pages of higher quality to the top of the results list. Hence, the search list becomes not only reduced in its size after the irrelevant Web-pages are excluded, but also has high-quality pages at the top of the list.

7.3 Summary

The aim of this section is to summarise the key findings of this research, as follows

- The Web is the richest source of digitally stored information; however, traditional approaches to Web-wide information gathering do not provide effective means to find relevant information based on the keywords match technique.
- Accessible information about the dynamically changing Real World is stored in the persistent Global Knowledge Base – the only available information that is attenuated to a great extent.

- Ashby's Law of Requisite Variety requires tools that are used for information gathering to provide enough amplification for the variety of the extracted information. Traditional keyword search approach does not comply with this law.
- Limited access to information similarly to the information overload will result in lower quality of decision making.
- Rumsfeld's philosophy on known knowns, known unknowns and unknown unknowns provides framework for representing uncertainty in total knowledge on a topic.
- A combination of keyword search and query-independent sorting algorithm contradicts the importance of relevance aspect in information search.
- Existing information gathering approaches lack one or more of the fundamental metrics for Web-based information extraction: relevance, completeness and quality. The proposed solution overcomes the limitations associated with these approaches and addresses all three metrics of the effective search model.
- Combination of statistical and semantic approaches resulted in the effective method to formulate a topic-focused collection of the Web content that is semantically related to the search topic.
- Conversion of "unknowns" into "knowns" is seen as a method of tracking the reduction of uncertainty during information discovery. Paired with Grounded Theory it provides a completeness metric for keeping the level of information within the boundaries for effective decision support
- Dempster Shafer theory provides a quality metric (Belief function) for valuating of discovered information that can be addressed from semantic view point.
- Results of the experiment proved the ineffectiveness of the traditional search engine to support effective decision making with up to one third of the search results turned out to be irrelevant to the search topic, with irrelevant Web-pages are often placed at the top of the results list and, in contrast, relevant quality pages positioned in the depth of the search results list.
- Relevance score probability density function indicates how well the corresponding query reflects the search topic in the eyes of the search engine allowing to adjust the scope or direction of the Web-search

The role of the human factor in finding relevant information is crucial. The proposed search method is developed to analyse the full amount of textual content despite its length, layout or reading age. It is also 20 times faster when comparing to the human processing.

Chapter 8

Future Research

8.1 Overview to Chapter

This chapter discusses possible future work that may improve the performance of the proposed search method. Section 8.2 will focus on the quality metric of the method and suggest other parameters that may be considered in order to assess quality of a Web page as a source of information in addition to the semantic value of the textual content. In section 8.3 Web-page content extraction techniques will be discussed that can replace the suggested AlchemyAPI tool and significantly reduce the time of text extraction form Web-pages. Section 8.4 will discuss how the stop-word list adjustment may affect the performance of the search model. Code optimisation and overall performance of the search model are discussed in the Section 8.5. Section 8.6 summarises this chapter.

8.2 Other Parameters for Quality Metric

This research used Dempster-Shafer Belief function as a quality metric that relies on the semantic closeness score obtained in the earlier stage on the search algorithm.

Research by (Zhu & Gauch, 2000) suggests an approach to calculate quality of a Web site on a per-topic basis by using six metrics. The following metrics are used: currency, availability, information-to-noise ratio, authority, popularity and cohesiveness. Currency is measured as the time stamp of the last modification of the document. Availability is calculated as the number of broken links on a page divided by the total numbers of links it contains. Information-to-noise ratio is computed as the total length of the tokens after pre-processing divided by the size of the document. Popularity score can be gained from the number of links pointing to a Web-page. Cohesiveness was determined by how closely related the major topics in the Web-page were. Authority of a Web-page can be measured with the equation (8.1), using age of domain (age_{domain}), number of links from other Web-

sites that point to the entire domain (N_{links}) and size of the Web-site that relates to the amount of quality information on the Web-site ($size_{website}$):

$$Authority = \log_{10}(age_{domain} \times N_{links} \times size_{website}) \quad (8.1)$$

The necessary Web-site statistics can be found with an available Web-site analysis tool.

Having obtained the metrics measurements, the quality of the site can be then determined by its information quality using the following equation:

$$G_i = \overline{W_i} * (a_s'' * \overline{T_i} + b_s'' * \overline{A_i} + c_s'' * \overline{I_i} + d_s'' * \overline{R_i} + e_s'' * \overline{P_i} + f_s'' * C_i) \quad (8.2)$$

where $\overline{W_i}$, $\overline{T_i}$, $\overline{A_i}$, $\overline{I_i}$, $\overline{R_i}$ and $\overline{P_i}$ are the means of information quantity, currency, availability, information-to-noise ratio, authority, and popularity of site *i* across topics relevant to the query. C_i is the cohesiveness of site I; a_s'', b_s'', c_s'', d_s'', e_s'' and f_s'' are the weights of each quality metric.

The Web-page quality metric can be further developed by considering some of the Web-site parameters that would allow assessing the Web-page as a source of information in addition to the semantic closeness score.

8.3 Accurate Web-page Content Extraction

This research relies on the AlchemyAPI tool to extract main textual content from the Web-pages. Experiments have shown that the majority of the running time is devoted to the text extraction part of the search algorithm. Moreover, the model testing resulted in the lower precision score (67%) for identifying content which is semantically relevant to the search topic. Additional techniques may need to be considered that would still provide accurate text extraction form Web-pages, but significantly reduce the extraction time.

The traditional approach to extracting meaningful text from a Web-page is based on deconstructing the HTML page to its Document Object Model (DOM) (Marini, 2002), as shown in figure 8.1.

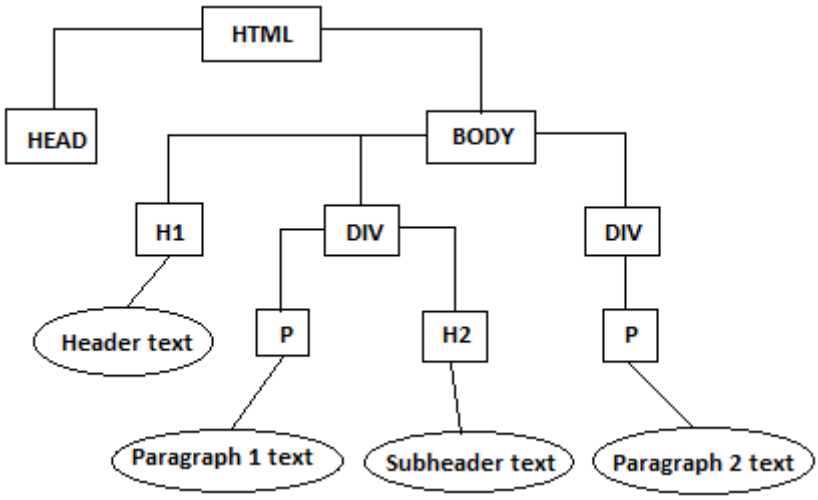


Figure 8.1 – Example Document Object Model of a Web page

An effective solution to consider in future research is based on deconstructing the HTML page to its DOM to detect and eliminate noise information from Web-pages was proposed by (Adam, Bouras, & Poulopoulos, 2009). Authors apply a set of algorithms in order to clean and correct the HTML code, locate and characterise each node of the DOM model and finally store the text from the nodes that are characterised as “useful” text nodes. Based on the deconstructed HTML page only paragraph “useful” texts will be extracted and processed.

Another Web-page “cleaning” technique proposed in (Yi, Liu, & Li, 2003) focuses on the commercial Web-sites and is based on the observation that in a typical Web site, Web pages tend to follow some fixed layouts or presentation styles as most pages are generated automatically. Those parts of a page whose layouts and actual contents (i.e., texts, images, links, etc) also appear in many other pages in the site are more likely to be noises, and those parts of a page whose layouts or actual contents are quite different from other pages are usually the main contents of the page. In their paper the authors first introduce style tree structure to capture the common layouts (or presentation styles) and the actual contents of the pages in a Web site. An information based measure is proposed to determine which parts of the style tree indicate noises and which parts of the style tree contain the main contents of the pages in the Web site. To clean a new page from the same site, the method simply maps the page to the style tree of the site. According to the mapping, the noisy parts are then detected and deleted.

The approach that may be also considered is presented in (Zhou, Xiong, & Liu, 2009); the authors claim that compared with existing DOM based approaches, the proposed solution is simple and fast with satisfied accuracy. Their solution is based on a practical observation that the main text of a Web page usually occupies the centre of the Web page with multiple adjacent, relatively long text paragraphs, especially for Web pages of news, articles, blogs, etc. When all pieces of visible text in a Web page are extracted, they are organised in paragraphs that are often located in the middle of the paragraph list with longer length.

8.4 Stop-Word List Adjustment

After the text is extracted, it is still loaded with so-called meaning-free parts of speech, or stop-words. Stop-word cannot characterise the text; and in the automatic identification of stop-word, the computer programs cannot tell whether one word characterises the text intelligently.

The search model proposed in this thesis uses on a common list of stop-words for any search topic. Future research may look at the changes (if any) in the efficiency and accuracy of the semantic test after the stop-words are reviewed and adjusted based on the search topic.

8.5 Code Optimisation

In the previous chapter it was estimated that the proposed solution is able to perform information gathering task 20 times faster comparing to the human processing. Semi-automated model was developed in Chapter 5 and uses manual processing combined with Python scripts and MS Excel statistical analysis and visualisation of results. If the algorithm's code gets optimised to provide more efficient use of the computing resources, the processing time can be significantly reduced.

8.6 Summary

This chapter discussed possible future work that may improve the performance and accuracy of the solution proposed in this research.

As such, the quality metric can be further developed by including additional Web-page parameters, such as currency, availability, information-to-noise ratio, authority, popularity and cohesiveness. Such combination would allow assessing the Web-page as a source of information in addition to the semantic closeness score.

Following the experiments, the text extraction from the Web-pages used the majority of the processing time. Also, the model testing resulted in the lower precision score (67%) for identifying content which is relevant to the defined search topic. This chapter discussed additional techniques to consider that would still provide accurate text extraction form Web-pages, but significantly reduce the extraction time. Among them were deconstructing the HTML page to its Document Object Model, mapping Web-pages to the Web-site style tree, or visual analysis of a Web pages text.

This chapter also suggested further exploration on how the adjustment of the stop-word list depending on the search topic may affect the performance of the search model.

Finally, more efficient use of the computing resources via code optimisation may significantly reduce the processing time.

Chapter 9

Conclusions and Summary

The main contribution of this research is that it addresses the issues associated with traditional information gathering and presents a novel semantic approach method to Web-based discovery of previously unknown intelligence for effective decision making. It provides a comprehensive theoretical background to the proposed solution together with a demonstration of the effectiveness of the method from results of the experiments, showing how the quality of collected information can be significantly enhanced by previously unknown information derived from the available known facts.

The objective of this research was:

> *To develop an* algorithm *that supports topic related search for Web-based information and estimates the quality of the extracted content, proving it is significantly enhanced comparing to the traditional Web-search approaches.*

This thesis introduced a new semantic Web-based search method that automates the correlation of topic-related content for discovery of hitherto unknown information from disparate and widely diverse Web-sources. This method is in contrast to traditional search methods that are constrained to specific or narrowly defined topics. It addresses the three key aspects of the information: semantic closeness to search topic, information completeness, and quality. The method is based on algorithms from Natural Language Processing combined with techniques adapted from grounded theory and Dempster-Shafer theory to significantly enhance the discovery of topic related Web-sourced intelligence.

In order to meet the research objective, a research method was developed and addressed throughout the chapters of this thesis.

In Chapter 2 the nature of information gathering process was analysed and issues associated with current Web-based search methods were identified. It was concluded that Web-content is ill structured, time related and substantially contradictory. Traditional Web search engines index Web-pages on keywords and present results ordered by in order of popularity and not by relevance. It also provided overview to the existing methods that overcome the issues associated with relevance, completeness and quality aspects of the

returned search results. This chapter also introduced some fundamental theories, such as Ashby's law of requisite variety and Rumsfeld's philosophy, that are usually ignored in the existing search methodologies. The proposed solution has been introduced and proven to have no limitations for Web-sources and open to perform search and analysis through any Web-page available to the search engine.

A comprehensive theoretical background for the novel search method was developed in Chapter 3. Three stages of filtering were suggested to narrow down and reorganise a search engine results. During the first stage the natural language processing methods filter only those pages that are semantically relevant to the search topic. The grounded theory part is the second stage that assesses the filtered Web-pages for holding new information and checks if this information exhausts the search topic. Lastly, the evidence on the Web-pages is analysed with Dempster-Shafer theory and its quality level is calculated as Belied function. The overall algorithm of the search model was presented at the end of this chapter.

The theoretical background is followed by the quantitative details for each stage of the algorithm in Chapter 4. A precise mathematical model has been created, supporting the theory behind the new algorithm. Mathematics of the semantic analysis explained how to select Web-pages that are relevant to the initial knowledge base content. Grounded theory is then used to identify the effect of the new information on the completeness of the initial knowledge base – for each iteration a conversion rate of "known unknowns" is used to analyse the change in new concepts. Main concepts of Dempster-Shafer theory were related to the context of Web-based discovery of unknowns.

Model design and software were addressed in Chapter 5. UML diagrams were developed to show the interaction between the objects. Pseudo-code was used to describe key pieces of code corresponding to all stages of the algorithm. Description was provided to all stages of the model: Web-Search and processing of Google results, pre-processing of extracted text and semantic closeness measurement stage, Grounded theory based completeness measurement, as well as measurement of quality score for semantic related content. Model testing was performed to assess how system generated results are compared with human manual analysis results. A random sample of Google search results was manually reviewed and assigned with either Relevant or Not Relevant tag. The testing results showed that system can recognise relevant content with precision 67% and recall 93%.

Chapter 6 described five experiments that had been chosen for the purpose of this research. First part of the experiments used search topics which were well defined with the corresponding queries focused around the initial knowledge base. The second part of the

experiments considered testing the system's performance when the topic understanding was incomplete and, thus, the chosen queries were remote from the search topic defined in the initial knowledge base text. A set of experiments carried out with different queries has proven that within the list of search engine results there is a large portion of Web-pages containing high-quality previously unknown information which would have not been identified using traditional methods. The experiment results were illustrated with various graphs and tables.

Results of this research were discussed in Chapter 7. It outlined key research assumptions and findings in accordance with the set research method, and provided research conclusions at the end of the chapter.

Possible future work in Chapter 8 suggested exploring techniques that would enhance the proposed quality metric, as well as accuracy of the semantic closeness score measurement to improve the precision score of the search model. It also suggested code optimisation for more efficient use of the computing resources, and other possible methods for the extraction of textual content on a Web-page in order to reduce the processing time.

This research has shown that in contrast to the current research, which tends to focus on specifically selected topics and have limited functionality, the new solution has no limitations for Web-sources and is open to perform search and analysis through any Web-page available to the search engine. The proposed new search method may be used to harvest Web-wide data in accordance with controlled parameters and subsequently transfer this data to a knowledge base where enterprise search technologies may be applied in the traditional way, providing rich and controlled information basis for future decision making.

Two papers have been produced relating to the research carried out on the topic of Web-based discovery of information for decision support:

- Danilova, N., & Stupples, D. (2012). *Application of Natural Language Processing to Web-based Intelligence Information Acquisition*. EISIC2012 Proceedings. Odense, Denmark: IEEE Computer Society.

- Danilova, N., & Stupples, D. (2013). *Semantic Approach to Web-based Discovery of Unknowns to Enhance Intelligence Gathering***.** International Journal of Information Retrieval, 3(1), pp. 24-42.

Bibliography

Ackoff, R. 2010. *Differences That Make a Difference: An Annotated Glossary of Distinctions Important in Management*. Axminster, Devon, UK: Triarchy Press Limite.

Adam, G., Bouras, C., & Poulopoulos, V. 2009. *CUTER: An efficient useful text extraction mechanism*. Advanced Information Networking and Applications Workshops (WAINA) (pp. 703-708. Institute of Electrical and Electronics Engineers (IEEE).

Agichtein, E., Castillo, C., Donato, D., A., G., & Gilad, M. 2008. *Finding High-Quality Content in Social Media*. Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM'08) (pp. 183-193. New York, NY, USA: ACM.

Ash, R. B., 1965. *Information Theory.* New York, NY: Interscience.

Ashby, W. R. 1956. *An Introduction to Cybernetics*. Chapman and Hall: University Paperback.

Ashton, J. 2009, February 1. *Autonomy is at the Heart of a New Data Revolution*. The Sunday Times, 1 February.

Autonomy. 2008, September. *Autonomy IDOL Structured Probabilistic Engine*. Retrieved September 23, 2012, from NDM Technologies Web site: http://www.ndm.net/archiving/pdf/hp/20090929_PI_PB_StructuredData_Web.pdf

Autonomy. 2009, September 29. *Autonomy Technology Overview*. Retrieved June 01, 2012, from Autonomy Corporation Web site: http://publications.autonomy.com/pdfs/Power/White%20Papers/Autonomy%20Technology/20090928_PI_WP_TechOverview_Web.pdf

Barker, J., 2004. *Invisible Web: What it is, Why it exists, How to find it, and its inherent ambiguity.* Berkeley, CA, USA,

BBC News. 2000, September 1. *FBI Arrests Man in Shares Hoax. Retrieved from BBC News* Web site: http://news.bbc.co.uk/1/hi/business/905315.stm

Berners-Lee, T. 2001, May 1. *The Semantic Web*. Scientific American.

Beynon-Davies, P., 2004. *Database Systems (3rd Edition)*. Palgrave, Basingstoke, UK:

Bird, S., Loper, E., & Klein, E. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Budanitsky, A., & Hirst, G. 2006, March. *Evaluating WordNet-based measures of semantic distance*. Computational Linguistics, 13-47.

Capiello, C., Daniel, F., Matera, M., & Pautasso, C. 2010. *Information quality in Mashups*. Internet Computing, 14(4), 14-22.

Carnap, R., & Bar-Hillel, Y. 1952. *An Outline of a Theory of Semantic Information*. Massachusetts Institute of Technology, Research Laboratory of Electronics, Cambridge, Massachusetts.

Chen, H., 2011. *Dark Web: Exploring and Data Mining the Dark Side of the Web.* :Springer Science & Business Media.

Chiang, M. 2012. *Networked Life: 20 Questions and Answers*. NY, USA: Cambrige University Press.

Cisco. 2014, June 10. *Cisco Visual Networking Index: Forecast and Methodology, 2013–2018*. Retrieved July 04, 2014, from Cisco Web site: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

Corbin, J., & Strauss, A. 2008. *Basics of qualitative research: techniques and procedures for developing grounded theory (3rd edition ed.).* London: Sage Publications.

Corley, C., & Mihalcea, R. 2005. *Measuring the semantic similarity of texts*. EMSEE '05 Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment (pp. 13-18. Stroudsburg, PA, USA: Association for Computational Linguistics.

Costa, P. & Laskey, K., 2006. *PR-OWL: A Framework for Probabilistic Ontologies.* pp. 237-249.

Grau, B. C. H., Motik, B., Parsia, B., Patel-Schneider, P., & Sattler, U. 2008. *OWL 2: The next step for OWL*. Journal of Web Semantics, 6(4), 309-322.

Grau, B. C., Motik, B., Stoilos, G., & Horroks, I. 2012. *Completeness guarantees for incomplete ontology reasoners: theory and practice*. Journal of Artificial Intelligence Research, 419-476.

Dagan, I., Lee, L., & Pereira, F. 1994. *Similarity-Based Estimation of Word*. 32nd annual meeting on Association for Computational Linguistics (ACL '94) (pp. 272-278. Las Cruces, New Mexico, USA: Association for Computational Linguistics.

Danilova, N., & Stupples, D. 2012. *Application of Natural Language Processing to Web-based Intelligence Information Acquisition*. European Intelligence and Security Informatics Conference (EISIC'2012. Odense, Denmark: IEEE Computer.

Deerwester, S. 1988. *Improving Information Retrieval with Latent Semantic Indexing*. Proceedings of the 51st Annual Meeting of the American Society for Information Science (ASIS'88), 25, pp. 36-40.

Delgado, J., Laplanche, R. & Krishnamurthy, V., 2005. *Bridging Structured and Unstructured Information.* The Information Management Journal, pp. 40-46.

DMOZ. 2014. *DMOZ Web Site*. Retrieved July 4, 2014, from About DMOZ: http://www.dmoz.org/docs/en/about.html

English, L., 2009. *Information Quality Applied: Best Practices for Improving Business*

Information, Processes and System. :Wiley.

Eppler, M., 2006. *Managing Information Quality: Increasing the Value of Information in Knowledge-intensive Products and Processes.* :Springer Science & Business Media.

Fellbaum, C. 1998. WordNet: an electronic lexical database. Cambridge, MA, USA: The MIT Press.

Fensel, D. et al., 2006. *Enabling Semantic Web Services: The Web Service Modeling Ontology.* :Springer Science & Business Media.

Fensel, D., 2005. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential.* :MIT Press.

Firth, J. 1957. A Synopsis of Linguistic Theory. Studies in Linguistic Analysis. Special volume of the Philological Society., 168-205.

Fischer, G., 2001. User Modeling in Human–Computer Interaction. *User Modeling and User-Adapted Interaction,* Volume 11(Issue 1-2), pp. 65-86.

Floridi, L. 2004, May. *Outline of a Theory of Strongly Semantic Information*. Minds and Machines, 14(2), 197-221.

Floridi, L. 2013, February 4. *Semantic Conception of Information*. Retrieved March 8, 2013, from Stanford Encyclopedia of Philosophy: http://plato.stanford.edu/entries/information-semantic/

Fox, C. J. 1990. *A stop list for general text*. ACM Special Interest Group on Information Retrieval Forum 24. pp. 19-35. ACM.

Fu, L., Meng, Y., Xia, Y., & Yu, H. 2010. *Web-content extraction based on Web-page layout analysis*. Second International Conference on Information Technology and Computer Science (ITCS 2010), (pp. 40-43. Ukraine.

Gao, Q., & Houben, G. 2010. *A Framework for Trust Establishment and Assessment on the Web of Data*. Proceedings of the 19th international conference on World Wide Web (WWW '10) (pp. 1097-1098. New York, NY, USA: ACM.

Gomes, B., & Smith, B. 2000. *Patent No. 684542*. USA.

Grau, B. & Stoilos, G., 2011. *What to ask to an incomplete semantic web reasoner?.* AAAI Press, pp. 2226-2231.

Grimes, C. 2010, June 8. *Google - Official Blog*. Retrieved July 04, 2014, from Our new search index: Caffeine: http://googleblog.blogspot.co.uk/2010/06/our-new-search-index-caffeine.html

Gruber, T. R. 1993. *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5(2), 199-220.

Gut, A., 2013. *Probability: A Graduate Course.* 2nd ed. New York, NY: Springer.

Hersh, W., 2008. *Information Retrieval: A Health and Biomedical Perspective.* :Springer Science & Business Media.

Hirst, G., & St-Onge, D. 1998. *Lexical chains as representations of context for the detection and correction of malapropisms*. In C. Fellbaum, WordNet: an Electronic Lexical Database (pp. 305-332. Cambridge, MA, USA: The MIT Press.

Hochstotter, N. & Lewandowski, D., 2009. *What Users See – Structures in Search Engine Results Page*s. Information Sciences, Volume 179.

Holzinger, A. et al., 2013. *Combining HCI, Natural Language Processing, and Knowledge Discovery - Potential of IBM Content Analytics as an assistive technology in the biomedical domain*. In: Springer Lecture Notes in Computer Science LNCS 7947. Heidelberg, Berlin, New York: Springer, pp. 13-24.

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. 2003. *From SHIQ and RDF to OWL: the making of a Web ontology language*. Journal Web Semantics, 1(1), 7-26.

Hu, G., & Zhao, Q. 2010. *Study to eliminating noisy information in Web-pages based on data mining*. Sixth International Conference on Natural Computation (ICNC 2010), 2, pp. 660-663.

Huutunen, S., Vihavainen, A., Du, M., & Yangarber, R. 2012. *Predicting Relevance of Event Extraction for the End User*. In Multi-source, Multilingual Information Extraction and Summarization (pp. 163-176. Berlin, Germany: Springer.

Jakus, G., Milutinovi, V., Omerovi, S. & Tomazi, S., 2013. *Concepts, Ontologies, and Knowledge Representation.* :Springer Science & Business Media.

Jerman-Blazic, B. 2010. *The Future of the Internet: Tussles and Challenges in the Evolution Path as Identified*. The Fourth International Conference on Digital Society (ICDS 2010) (pp. 25-30. St. Maarten, Netherlands: IEEE.

Jiang, J., & Conrath, D. 1997. *Semantic similarity based on corpus statistics and lexical taxonomy*. International Conference on Research on Computational Linguistics. Taipei, Taiwan.

Kao, J., Lin, S., Ho, J., & Chen, M. 2002. *Entropy-based link analysis for mining Web informative structures*. Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM'2002) (pp. 574-581. ACM Press.

Kopcke, H., Thor, A., & Rahm, E. 2010. *Learning-based Approaches for Matching Web Data Entitles*. Internet Computing, 14(4), 23-31.

Kullback, S. 1997. *Information theory and statistics*. Mineola, NY, USA: Courier Dover Publications.

Langville, A. N., & Meye, C. D. 2009. *Google's PageRank and Beyond: The Science of Search Engine Ranking*. Princeton, NJ, USA: Princeton University Press.

Laskey, K. 2008, February. *MEBN: A language for first-order Bayesian knowledge bases*.

Artificial Intelligence(172 (2-3)), 140-178.

Laskey, K. B., Schum, D. A., Costa, P. C., & Janssen, T. 2008. *Ontology of Evidence*. Proceedings of the Third International Ontology for the Intelligence Community Conference (OIC 2008. Fairfax, VA, USA: CEUR Workshop Proceedings.

Laskey, K., Costa, P., & Janssen, T. 2008. *Probabilistic ontologies for knowledge fusion*. 11th International Conference on Information Fusion, (pp. 1-8. Cologne, Germany.

Leacock, C., & Chodorow, M. 1998. *Combining local context and WordNet similarity for word sense identification*. In C. Fellbaum, WordNet: An Electronic Lexical Database (pp. 265-283. Cambridge, MA, USA: MIT Press.

Lee, L. 1999. *Measures of Distributional Similarity*. 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL'99) (pp. 25–32. College Park, Maryland, USA: Association for Computational Linguistics.

Lee, L. 2001. *On the Effectiveness of the Skew Divergence for Statistical Language Analysis*. Artificial Intelligence and Statistics (AISTAT'01) (pp. 65-72. Key West, FL, USA: Morgan Kaufmann Publishers.

Lehmann, J., Schüppel, J., & Auer, S. 2007. *Discovering Unknown Connections - the DBpedia Relationship Finder*. Proceedings of the 1st SABRE Conference on Social Semantic Web (CSSWeb'07), (pp. 99-110. Leipzig, Germany.

Levering, R. & Cutler, M., 2006. *The Portrait of a Common HTML Web Page.* Amsterdam, The Netherlands, pp. 198-204.

Li, B. et al., 2011. *An Analysis of Anonymity Usage.* Vienna, Austria, Springer-Verlag, p. 113–116.

Lovins, J. B., 1968. *Development of a Stemming Algorithm*. Mechanical Translation and Computational Linguistics, Volume 11, p. 22–31.

Luhn, H. 1958. *The automatic creation of literature abstracts*. IBM Journal, 159-165.

Madhavan, J. et al., 2008. *Google's Deep-Web Crawl*. VLDB Endowment, pp. 1241-1252.

Marini, J. 2002. *Document Object Model: Processing Structured Documents*. McGraw-Hill Prof Med/Tech.

Marsha, A., 2009. *Web Wisdom: How To Evaluate and Create Information Quality on the Web, Second Edition.* :CRC Press.

Martin, P., & Turner, B. 1986. *Grounded theory and organizational research*. The Journal of Applied Behavioral Science , 22 (2), 141-157.

McGilvray, D., 2010. *Executing Data Quality Projects: Ten Steps to Quality Data and Trusted Information.* :Morgan Kaufmann.

Michalewicz, Z. 2007. *Adaptive Business Intelligence*. Springer.

Middleton, S. E., De Roure, D., & Shadbolt, N. 2001. *Capturing knowledge of user preferences: ontologies in recommender systems*. Proceedings of the 1st international conference on Knowledge capture (K-CAP '01) (pp. 100-107. New York, NY, USA: ACM.

Middleton, S. E., Shadbolt, N., & De Roure, D. 2004, January. Ontological user profiling in recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 54-88.

Middleton, S., De Roure, D., & Shadbolt, N. 2002. Foxtrot Recommender System: User profiling, Ontologies and the World Wide Web. 11th International World Wide Web Conference (WWW2002. Hawaii, USA.

Mihalcea, M., Corley, C. & Strapparava, C., 2006. Corpus-based and knowledge-based measures of text semantic similarity. pp. 775-780.

Miller, G., & Charle, W. 1991. Contextual correlates of semantic similarity. Language and Cognitive Processes, 6(1), 1-28.

Mohammad, S. 2008. Measuring Semantic Distance Using Distributional Profiles of Concepts (PhD Thesis). Toronto, Canada: University of Toronto.

Mohammad, S., & Hirst, G. 2006. Measuring semantic distance, using distributional profiles of concepts. Computational Linguistics, 1(1), 1-33.

Monk, E. & Wagner, B., 2009. Course Technology Cengage Learning. In: Concepts in Enterprise Resource Planning. Boston, MA, USA:

Morville, P. & Rosenfeld, L., 2006. Information Architecture for the World Wide Web: Designing Large-Scale Web Sites. :"O'Reilly Media, Inc.".

Netcraft. 2010, July. July 2010 Web Server Survey. Retrieved July 15, 2010, from Netcraft Internet services company: http://news.netcraft.com/archives/2010/07/16/july-2010-Web-server-survey-16.html

Netcraft. 2012, January. January 2012 Web Server Survey. Retrieved January 6, 2012, from Netcraft Internet services company: http://news.netcraft.com/archives/2012/01/03/january-2012-Web-server-survey.html

Netcraft. 2014, June. June 2014 Web Server Survey. Retrieved July 03, 2014, from Netcraft Internet services company: http://news.netcraft.com/archives/2014/06/06/june-2014-Web-server-survey.html

Prud'hommeaux, E., & Seaborne, A. 2008, January 15. SPARQL Query Language for RDF. Retrieved September 15, 2012, from W3C Recommendation: http://www.w3.org/TR/rdf-sparql-query/

Pugh, W., & Henzinger, M. 2001. Patent No. 768947. USA.

Raghavan, S. & Garcia-Molina, H., 2001. Crawling the Hidden Web. Rome, Italy,

Raiber, F., & KURLand, O. 2013. Using Document-Quality Measures to Predict Web-Search Effectiveness. 35th European Conference on IR Research (ECIR 2013) (pp. 134-145.

Moscow, Russia: Springer Berlin Heidelberg.

Rajaraman, A. & Ullman, J. D., 2011. Mining of Massive Datasets. :Cambridge University Press.

RDF Working Group. 2004, February 10. RDF Semantics. Retrieved September 15, 2012, from World Wide Web Consortium (W3C): http://www.w3.org/RDF/

Resnik, P. 1995. Using information content to evaluate semantic similarity. 14th international joint conference on Artificial intelligence (IJCAI'95. 1, pp. 448-453. Montreal, Canada: Morgan Kaufmann Publishers Inc.

Roget, P. 2004, April 19. Roget's Thesaurus of English Words and Phrases. Retrieved from Project Gutenberg: http://www.gutenberg.org/ebooks/10681

Roulston, M. S., 1999. Estimating the errors on measured entropy and mutual information. Physica D: Nonlinear Phenomena, 125(3-4), pp. 285-294.

Rubenstein, H., & Goodenough, J. 1965. Contextual correlates of synonymy. Communications of the ACM , 8 (10), 627 -633.

Rumsfeld, D. 2002. News transcript: DoD news briefing. Washington D.C: U.S.Department of Defence.

Schum, D. A. 2001. The evidential foundations of probabilistic reasoning. Northwestern University Press.

Shafer, G. 1976. A mathematical theory of evidence. Princeton, USA: Princeton University Press.

Shannon, C. E. 1948. A Mathematical Theory of Communication. The Bell System Technical Journal, 27, 379-423.

Shikha, M., Hema, B. & Puman, B., 2009. BDI Planning Approach to Distributed Multiagent Based Semanti Seacrh Engine. Salamanca, Spain,

SMART. 1974. User's Manual for the SMART Information Retrieval System. Cornell University.

Smartlogic. 2001. Semaphore Semantic Enhancement - A Smartlogic Fact Sheet. Retrieved May 14, 2011, from Smartlogic Web site: http://www.smartlogic.com/request-download?f=35/SL_SES_WPNT0012011_EN.pdf

Spyrou, E., Iakovidis, D. & Mylonas, P., 2014. Semantic Multimedia Analysis and Processing. :CRC Press.

Stevenson, M. & Wilks, Y., 2003. Word-sense disambiguation. In: R. Mitkov, ed. The Oxford Handbook of Computational Linguistics. :Oxford University Press, pp. 249-265.

Stoilos, G., Grau, B. C., & Horrocks, I. 2010. How Incomplete is your Semantic Web Reasoner? Proceedings of the 24th National Conference on Artificial Intelligence (AAAI 10) (pp. 1431-1436. AAAI Publications.

Stoilos, G., Grau, B. C., & Horroks, I. 2010. Completeness Guarantees for Incomplete Reasoners. 9th International Semantic Web Conference (ISWC 2010) (pp. 747-763. Shanghai, China: Springer Berlin Heidelberg.

The Economist, 2010. Data, Data Everywhere. A Special Report on Managing Information.. The Economist, 25 February.

The World Wide Web Consortium (W3C). 2009, November 12. OWL Web Ontology Language Reference. Retrieved September 23, 2012, from W3C: http://www.w3.org/TR/owl-ref/

Torisawa, K., De Saeger, S., Kazama, J., Sumida, A., Noguchi, D., Kakizawa, Y., et al. 2010. Organizing the Web's Information Explosion to Discover Unknown Unknowns. New Generation Computing, 28(3), 217-236.

van Rijsbergen, C. J. 1979. Information retrieval. London: Butterworths.

Vargas, L. 2008, October 22. Mike Lynch, CEO and founder of Autonomy. Retrieved January 2, 2012, from Business Weekly: http://www.businessweekly.co.uk/q10-talking-heads/11508-mike-lynch-ceo-and-founder-of-autonomy

Wai-Ting Leung, K., Ng, W. & Lun Lee, D., 2008. Personalized Concept-Based Clustering of Search Engine Queries. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 20(11), pp. 1505-1518.

Ward Church, K., & Hanks, P. 1990, March. Word association norms, mutual information, and lexicography. Computational Linguistics, 16(1), 22–29.

Yang, C., Chen, H., & Hong, K. 2003. Visualization of large category map for internet browsing. Decision Support Systems, 35(1), 89-102.

Yeung, R. W., 2008. The Science of Information. Information Theory and Network Coding, p. 1–10.

Yi, L., Liu, B., & Li, X. 2009. Eliminating noisy information in Web-pages for data mining. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 296-305. New York, USA: ACM.

Zailani, S. & Rajagopal, P., 2007. Information Quality Management: Theory and Applications. :Idea Group Inc (IGI).

Zhou, B., Xiong, Y., & Liu, W. 2009. Efficient Web-page main text extraction towards online news analysis. International Conference on e-Business Engineering, (ICEBE '09) pp. 37-41. IEEE.

Zhou, Y., Cuenca Grau, B. & Horrocks, I., 2012. Efficient Upper Bound Computation of Query Answers in Expressive Description Logics.

Appendix 1 – Initial Knowledge Base Text

Search topic “tobacco industry”

The tobacco industry comprises those persons and companies engaged in the growth, preparation for sale, shipment, advertisement, and distribution of tobacco and tobacco-related products. It is a global industry; tobacco can grow in any warm, moist environment, which means it can be farmed on all continents except Antarctica.

Tobacco is a commodity product similar in economic terms to foodstuffs in that the price is set by the fact that crop yields vary depending on local weather conditions. The price varies by specific species grown, the total quantity on the market ready for sale, the area where it was grown, the health of the plants, and other characteristics individual to product quality. Laws around the world now often have some restrictions on smoking but, still 5.5 trillion cigarettes are smoked each year. Taxes are often heavily imposed on tobacco.

The tobacco industry generally refers to the companies involved in the manufacture of cigarettes, cigars, snuff and chewing and pipe tobacco. The largest tobacco company in the world by volume is China National Tobacco Co. Following extensive merger activity in the 1990s and 2000s, international markets are dominated by five firms: Philip Morris International, British American Tobacco, Japan Tobacco, Altria, and Imperial Tobacco.

The tobacco industry in the United States has suffered greatly since the mid-1990s, when it was successfully sued by several U.S. states. The suits claimed that tobacco causes cancer, that companies in the industry knew this, and that they deliberately understated the significance of their findings, contributing to the illness and death of many citizens in those states.

Lawsuits against the tobacco industry are primarily restricted to the United States due to differences in legal systems in other countries. Many businesses class ongoing lawsuits as a cost of doing business in the US and feel their revenue will be only marginally affected by the activities.

Search topic “cocaine smuggling”

Colombia is Cocaine’s Main Producer

Cocaine is produced from the leaves of the coca plant (Erythoxylon coca). The plant grows almost exclusively in northern and western South America. Colombia is now the main producer of illegal cocaine with Peru, Bolivia, and Chile providing significant amounts of the drug. The coca plant grows best in the mountain and jungle areas of these countries.

Small cocaine brick and press

Small laboratories are scattered throughout the coca growing areas of South America. The hand picked coca leaves are soaked in gasoline and other chemicals to extract the coca base from the leaves in industrial-sized drums. Then the base is poured into brick molds. The water is pressed out, leaving a hard, easy-to-handle brick containing about 50 percent cocaine. The bricks are sent to collection points where they are shipped to markets in the U.S. and other countries.

Colombians Control the Trade

Drug cartels based in Colombia control almost all cocaine trafficking. They process coca from Peru, Bolivia, and Colombia. Various cocaine trafficking organizations then ship it to all corners of the globe.

Cocaine Processing Labs

In these remote areas, processing laboratories are set up to extract the coca from the leaves and then convert the coca into cocaine. There are two types of processing labs: pozo pit labs that use acidic solutions, and the more common lab that uses metal drums and gasoline.

Step One

Cocaine is typically processed in jungle labs by first soaking leaves in gasoline inside metal drums.

Step Two

Gasoline containing cocaine alkaloid is drained from metal drums and filtered into a barrel with diluted acid. The gasoline will be removed from the acid layer and sodium bicarbonate or ammonia will be added to the solution to make cocaine base.

Step Three

Cocaine base is filtered through a cloth.

Step Four

The remaining substance is dried, resulting in a purer form of cocaine base.

Step Five

Cocaine base is dissolved in a solvent such as ethyl acetate, acetone, or ether and then heated in a hot water bath called a “bańo maria.” Another solvent such as methyl ethyl ketone is added to the boiling liquid along with concentrated hydrochloric acid which results in the crystallization of cocaine hydrochloride.

Step Six

Excess solvents are removed from the cocaine hydrochloride, first by hand, and then using a hydraulic press.

Step Seven

Solvents are finally removed using microwave ovens creating the basis for powder cocaine.

Step Eight

Cocaine base, commonly called "crack," is typically made by dissolving cocaine hydrochloride in a mixture of water and baking soda. The solution is boiled until the cocaine forms an oily substance that drops out of the solution and settles at the bottom of the container.

Step Nine

Excess water, sodium chloride, and other impurities can be removed after cooling the oil to a rock-like substance. The resulting product is crack cocaine.

Search topic “coffee production”

Coffee production is the industrial process of converting the raw fruit of the coffee plant into the finished coffee. The cherry has the fruit or pulp removed leaving the seed or bean which is then dried. While all green coffee is processed, the method that is used varies and can have a significant effect on the flavor of roasted and brewed coffee.

A coffee plant usually starts to produce flowers 3–4 years after it is planted,[1] and it is from these flowers that the fruits of the plant (commonly known as coffee cherries) appear, with the first useful harvest possible around 5 years after planting. The cherries ripen around eight months after the emergence of the flower, by changing color from green to red, and it is at this time that they should be harvested.[2] In most coffee-growing countries, there is one major harvest a year; though in countries like Colombia, where there are two flowerings a year, there is a main and secondary crop.[3]

In most countries, the coffee crop is picked by hand, a labor-intensive and difficult process, though in places like Brazil, where the landscape is relatively flat and the coffee fields immense, the process has been mechanized.[3] Whether picked by hand or by machine, all coffee is harvested in one of two ways:

Strip Picked: The entire crop is harvested at one time. This can either be done by machine or by hand. In either case, all of the cherries are stripped off the branch at one time.

Selectively Picked: Only the ripe cherries are harvested and they are picked individually by hand. Pickers rotate among the trees every 8–10 days, choosing only the cherries which are at the peak of ripeness. Because this kind of harvest is labor intensive, and thus more costly, it is used primarily to harvest the finer arabica beans.[3]

The laborers who pick coffee by hand receive payment by the basketful. As of 2003, payment per basket is between US$2.00 to $10 with the overwhelming majority of the laborers receiving payment at the lower end. An experienced coffee picker can collect up to 6-7 baskets a day. Depending on the grower, coffee pickers are sometimes specifically instructed to not pick green coffee berries since the seeds in the berries are not fully formed or mature. This discernment typically only occurs with growers who harvest for higher end/specialty coffee where the pickers are paid better for their labor. Mixes of green and red berries, or just green berries, are used to produce cheaper mass consumer coffee beans, which are characterized by a displeasingly bitter/astringent flavor and a sharp odor. Red berries, with their higher aromatic oil and lower organic acid content, are more fragrant, smooth, and mellow. As such, coffee picking is one of the most important stages in coffee production.

In the wet process, the fruit covering the seeds/beans is removed before they are dried. Coffee processed by the wet method is called wet processed or washed coffee.[4] The wet method requires the use of specific equipment and substantial quantities of water.

The coffee cherries are sorted by immersion in water.Bad or unripe fruit will float and the good ripe fruit will sink. The skin of the cherry and some of the pulp is removed by pressing the fruit by machine in water through a screen. The bean will still have a significant amount of the pulp clinging to it that needs to be removed. This is done either by the classic ferment-and-wash method or a newer procedure variously called machine-assisted wet processing, aquapulping or mechanical demucilaging:

In the ferment-and-wash method of wet processing, the remainder of the pulp is removed by breaking down the cellulose by fermenting the beans with microbes and then washing them with large amounts of water. Fermentation can be done with extra water or, in "Dry Fermentation", in the fruit's own juices only.

The fermentation process has to be carefully monitored to ensure that the coffee doesn't acquire undesirable, sour flavors. For most coffees, mucilage removal through fermentation takes between 24 and 36 hours, depending on the temperature, thickness of the mucilage layer and concentration of the enzymes. The end of the fermentation is assessed by feel, as the parchment surrounding the beans loses its slimy texture and acquires a rougher "pebbly" feel. When the fermentation is complete, the coffee is thoroughly washed with clean water in tanks or in special washing machines.[5]

In machine-assisted wet processing, fermentation is not used to separate the bean from the remainder of the pulp; rather, this is done through mechanical scrubbing. This process can cut down on water use and pollution since ferment and wash water stinks. In addition, removing mucilage by machine is easier and more predictable than removing it by fermenting and washing. However, by eliminating the fermentation step and prematurely separating fruit and bean, mechanical demucilaging can remove an important tool that mill operators have of influencing coffee flavor. Furthermore, the ecological criticism of the ferment-and-wash method increasingly has become moot, since a combination of low-water equipment plus settling tanks allows conscientious mill operators to carry out fermentation with limited pollution.[4]

Any wet processing of coffee produces coffee wastewater which can be a pollutant.[6] Ecologically sensitive farms reprocess the wastewater along with the shell and mucilage as compost to be used in soil fertilization programs. The amount of water used in processing can vary, but most often is used in a 1 to 1 ratio.

After the pulp has been removed what is left is the bean surrounded by two additional layers, the silver skin and the parchment. The beans must be dried to a water content of about 10% before they are stable. Coffee beans can be dried in the sun or by machine but in most cases it is dried in the sun to 12-13% moisture and brought down to 10% by machine. Drying entirely by machine is normally only done where space is at a premium or the humidity is too high for the beans to dry before mildewing.

When dried in the sun coffee is most often spread out in rows on large patios where it needs to be raked every six hours to promote even drying and prevent the growth of mildew. Some coffee is dried on large raised tables where the coffee is turned by hand. Drying coffee this way has the advantage of allowing air to circulate better around the beans promoting more even drying but increases cost and labor significantly.

After the drying process (in the sun and/or through machines), the parchment skin or pergamino is thoroughly dry and crumbly, and easily removed in the Hulling process. Coffee occasionally is sold and shipped in parchment or en pergamino, but most often a machine called a huller is used to crunch off the parchment skin before the beans are shipped.[4]

Dry process

Dry process, also known as unwashed or natural coffee, is the oldest method of processing coffee. The entire cherry after harvest is first cleaned and then placed in the sun to dry on tables or in thin layers on patios:[5]

The harvested cherries are usually sorted and cleaned, to separate the unripe, overripe and damaged cherries and to remove dirt, soil, twigs and leaves. This can be done by winnowing, which is commonly done by hand, using a large sieve. Any unwanted cherries or other material not winnowed away can be picked out from the top of the sieve. The ripe cherries can also be separated by flotation in washing channels close to the drying areas.

The coffee cherries are spread out in the sun, either on large concrete or brick patios or on matting raised to waist height on trestles. As the cherries dry, they are raked or turned by hand to ensure even drying and prevent mildew. It may take up to 4 weeks before the cherries are dried to the optimum moisture content, depending on the weather conditions. On larger plantations, machine-drying is sometimes used to speed up the process after the coffee has been pre-dried in the sun for a few days.

The drying operation is the most important stage of the process, since it affects the final quality of the green coffee. A coffee that has been overdried will become brittle and produce too many broken beans during hulling (broken beans are considered defective beans). Coffee that has not been dried sufficiently will be too moist and prone to rapid deterioration caused by the attack of fungi and bacteria.

The dried cherries are stored in bulk in special silos until they are sent to the mill where hulling, sorting, grading and bagging take place. All the outer layers of the dried cherry are removed in one step by the hulling machine.

The dry method is used for about 90% of the Arabica coffee produced in Brazil, most of the coffees produced in Ethiopia, Haiti and Paraguay, as well as for some Arabicas produced in India and Ecuador. Almost all Robustas are processed by this method. It is not practical in very rainy regions, where the humidity of the atmosphere is too high or where it rains frequently during harvesting.[5]

Semi dry process

Semi dry is a hybrid process used in Indonesia and Brazil. In Indonesia, the process is also called "wet hulled", "semi-washed" or "Giling Basah". Literally translated from Indonesian, Giling Basah means "wet grinding".[7] This process is said to reduce acidity and increase body.[8]

Most small-scale farmers in Sumatra, Sulawesi, Flores and Papua use the giling basah process. In this process, farmers remove the outer skin from the cherries mechanically, using locally built pulping machines. The coffee beans, still coated with mucilage, are then stored for up to a day. Following this waiting period, the mucilage is washed off and the parchment coffee is partially dried in the sun before sale at 30% to 35% moisture content.[8]

Milling

Structure of coffee berry and beans: 1: center cut 2:bean (endosperm) 3: silver skin (testa, epidermis), 4: parchment (hull, endocarp) 5: pectin layer 6: pulp (mesocarp) 7: outer skin (pericarp, exocarp)

The final steps in coffee processing involve removing the last layers of dry skin and remaining fruit residue from the now dry coffee, and cleaning and sorting it. These steps are often called dry milling to distinguish them from the steps that take place before drying, which collectively are called wet milling.[3] [4]

Hulling

The first step in dry milling is the removal of what is left of the fruit from the bean, whether it is the crumbly parchment skin of wet-processed coffee,the parchment skin and dried mucilage of semi-dry-processed coffee, or the entire dry, leathery fruit covering of the dry-processed coffee. Hulling is done with the help of machines, which can range from simple millstones to sophisticated machines that gently whack at the coffee.[3]

Polishing

This is an optional process in which any silver skin that remains on the beans after hulling is removed in a polishing machine.[3] This is done to improve the appearance of green coffee beans and eliminate a byproduct of roasting called chaff. It is described by some to be detrimental to the taste by raising the temperature of the bean through friction which changes the chemical makeup of the bean.

Cleaning and sorting

Most fine coffee goes through a battery of machines that sort the coffee by density of bean and by bean size, all the while removing sticks, rocks, nails, and miscellaneous debris that may have become mixed with the coffee during drying. First machines blow the beans into the air; those that fall into bins closest to the air source are heaviest and biggest; the lightest (and likely defective) beans plus chaff are blown in the farthest bin. Other machines shake the beans through a series of sieves, sorting them by size. Finally, a machine called a gravity separator shakes the sized beans on a tilted table, so that the heaviest, densest and best vibrate to one side of the pulsating table, and the lightest to the other.[4] [9]

The final step in the cleaning and sorting procedure is called color sorting, or separating defective beans from sound beans on the basis of color rather than density or size. Color sorting is the trickiest and perhaps most important of all the steps in sorting and cleaning. With most high-quality coffees color sorting is done in the simplest possible way: by hand. Teams of workers pick discolored and other defective beans from the sounds beans. The very best coffees may be hand-cleaned twice (double picked) or even three times (triple picked). Coffee that has been cleaned by hand is usually called European preparation; most specialty coffees have been cleaned and sorted in this way.[4]

Color sorting can also be done by machines. Streams of beans fall rapidly, one at a time, past sensors that are set according to parameters that identify defective beans by value (dark to light) or by color. A tiny, decisive puff of compressed air pops each defective bean out of the stream of sound beans the instant the machine detects an anomaly. However, these machines are currently not used widely in the coffee industry for two reasons. First, the capital investment to install these delicate machines and the technical support to maintain them is daunting. Second, sorting coffee by hand supplies much-needed work for the small rural communities that often cluster around coffee mills. Nevertheless, computerized color sorters are essential to coffee industries in regions with relatively high standards of living and high wage demands.[4]

Grading

Grading is the process of categorizing coffee beans on the basis of various criteria such as size of the bean, where and at what altitude it was grown, how it was prepared and picked, and how good it tastes, or its cup quality. Coffees also may be graded by the number of imperfections (defective and broken beans, pebbles, sticks, etc.) per sample. For the finest coffees, origin of the beans (farm or estate, region, cooperative) is especially important. Growers of premium estate or cooperative coffees may impose a level of quality control that goes well beyond conventionally defined grading criteria,

because they want their coffee to command the higher price that goes with recognition and consistent quality.

All coffee, when it was introduced in Europe, came from the port of Mocha in what is now modern day Yemen. To import the beans to Europe the coffee was on boats for a long sea voyage around the Horn of Africa. This long journey and the exposure to the sea air changed the coffee's flavor. Later, coffee spread to India and Indonesia but still required a long sea voyage. Once the Suez Canal was opened the travel time to Europe was greatly reduced and coffee whose flavor had not changed due to a long sea voyage began arriving. To some degree, this fresher coffee was rejected because Europeans had developed a taste for the changes that were brought on by the long sea voyage.[citation needed] To meet this desire, some coffee was aged in large open-sided warehouses at port for six or more months in an attempt to simulate the effects of a long sea voyage before it was shipped to Europe.

Although it is still widely debated, certain types of green coffee are believed to improve with age; especially those that are valued for their low acidity, such as coffees from Indonesia or India. Several of these coffee producers sell coffee beans that have been aged for as long as 3 years, with some as long as 8 years. However, most coffee experts agree that a green coffee peaks in flavor and freshness within one year of harvest, because over-aged coffee beans will lose much of their essential oil content.

Decaffeination is the process of extracting caffeine from green coffee beans prior to roasting. The most common decaffeination process used in the United States is supercritical carbon dioxide (CO2) extraction. In this process, moistened green coffee beans are contacted with large quantities of supercritical CO2 (CO2 maintained at a pressure of about 4,000 pounds force per square inch (28 MPa) and temperatures between 90 and 100 °C (194 and 212 °F)), which removes about 97% of the caffeine from the beans. The caffeine is then recovered from the CO2, typically using an activated carbon adsorption system.

Another commonly used method is solvent extraction, typically using oil (extracted from roasted coffee) or ethyl acetate as a solvent. In this process, solvent is added to moistened green coffee beans to extract most of the caffeine from the beans. After the beans are removed from the solvent, they are steam-stripped to remove any residual solvent. The caffeine is then recovered from the solvent, and the solvent is re-used. The Swiss Water Process is also used for decaffeination. Decaffeinated coffee beans have a residual caffeine content of about 0.1% on a dry basis. Not all facilities have decaffeination operations, and decaffeinated green coffee beans are purchased by many facilities that produce decaffeinated coffee.

Green coffee is usually transported in jute bags. While green coffee may be usable for several years, it is vulnerable to quality degradation based on how it is stored. Jute bags are extremely porous, exposing the coffee to whatever elements it is surrounded by. Coffee that is poorly stored may develop a bURLap-like taste known as "bagginess", and its positive qualities may fade.[10]

In recent years, the specialty coffee market has begun to utilize enhanced storage method. A gas barrier liner to jute bags, is sometimes used to preserve the quality of green coffee.[11] Less frequently, green coffee is stored in vacuum packaging; while vacuum packs further reduce the ability of green coffee to interact with oxygen at atmospheric moisture, it is a significantly more expensive storage option.[12]

Roasting

Main article: Coffee roasting

See also: Home roasting coffee

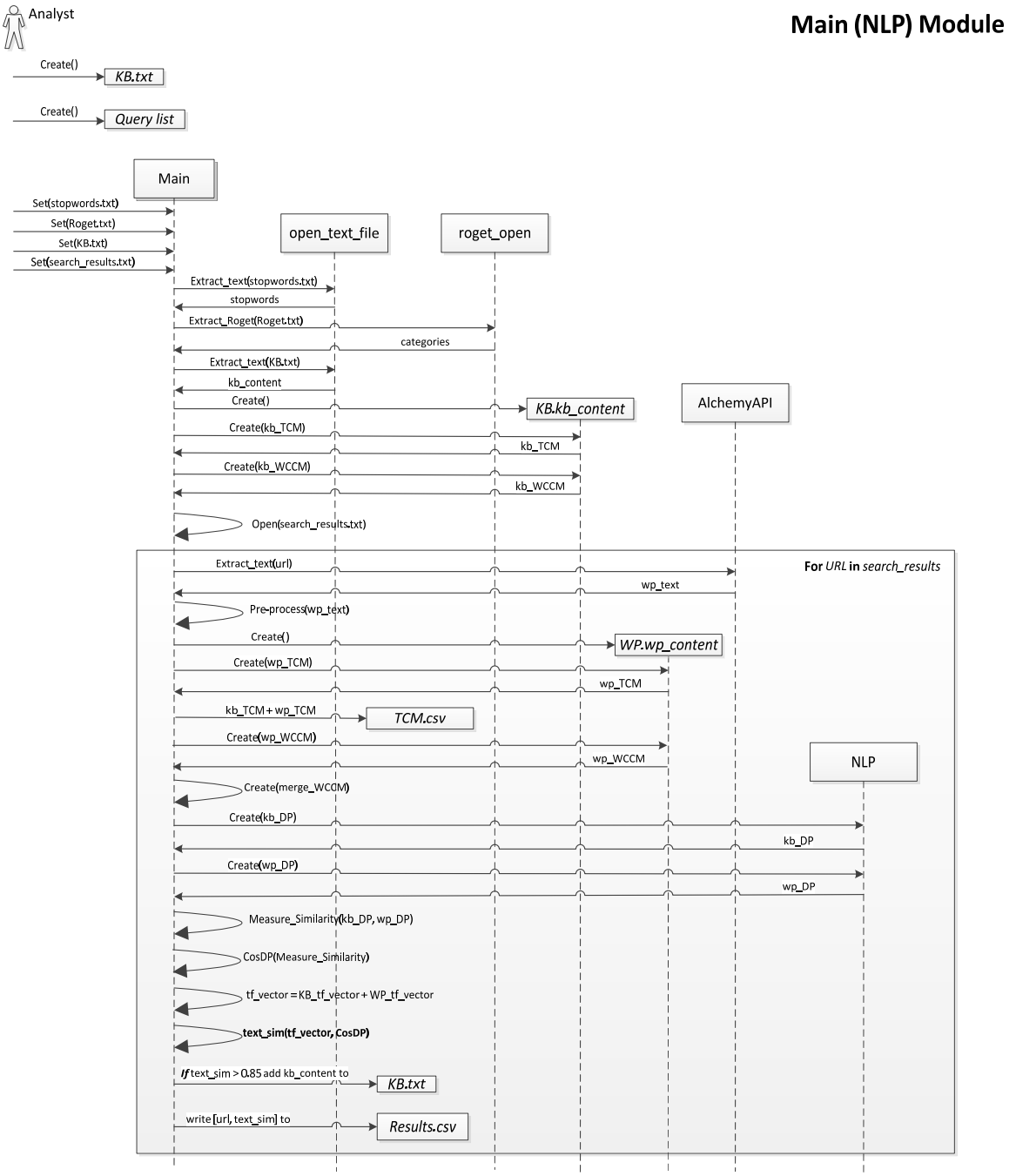
Although not considered part of the processing pipeline proper, nearly all coffee sold to consumers throughout the world is sold as roasted coffee in general one of four degrees of roasting: light, medium, medium-dark, and dark.[13] Consumers can also elect to buy unroasted coffee to be roasted at home.

Appendix 2 – Stop Word List

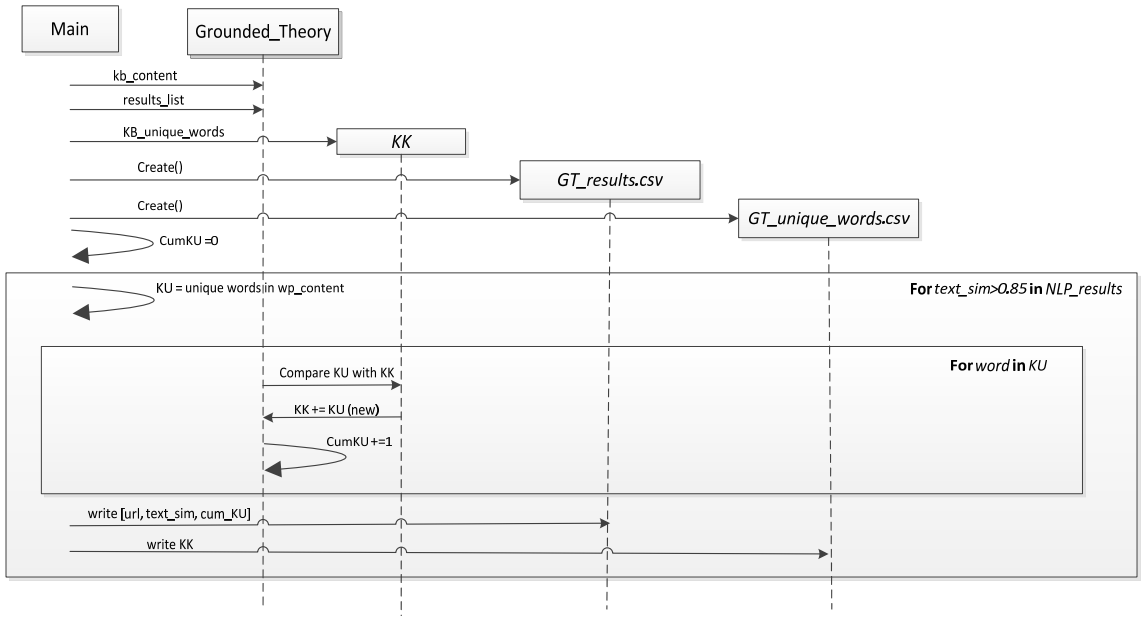
a	both	hasn't	make	out	there're	were
about	but	have	makes	over	there's	weren't
above	by	haven't	many	overall	there've	what
according	can	he	maybe	own	thereafter	what'll
across	can't	he'd	me	per	thereby	what's
actually	cannot	he'll	meantime	perhaps	therefore	what've
adj	caption	he's	meanwhile	rather	therein	whatever
after	co	hence	might	recent	thereupon	when
afterwards	company	her	million	recently	these	whence
again	corp	here	miss	same	they	whenever
against	corporation	here's	more	seem	they'd	where
all	could	hereafter	moreover	seemed	they'll	where's
almost	couldn't	hereby	most	seeming	they're	whereafter
alone	did	herein	mostly	seems	they've	whereas
along	didn't	hereupon	mr	seven	thirty	whereby
already	do	hers	mrs	seventy	this	wherein
also	does	herself	much	several	those	whereupon
although	doesn't	him	must	she	though	wherever
always	don't	himself	my	she'd	thousand	whether
among	down	his	myself	she'll	three	which
amongst	during	how	namely	she's	through	while
an	each	however	neither	should	throughout	whither
and	eg	hundred	never	shouldn't	thru	who
another	eight	i	nevertheless	since	thus	who'd
any	eighty	i'd	next	six	to	who'll
anyhow	either	i'll	nine	sixty	together	who's
anyone	else	i'm	ninety	so	too	whoever
anything	elsewhere	i've	no	some	toward	whole
anywhere	end	ie	nobody	somehow	towards	whom
are	ending	if	none	someone	trillion	whomever
aren't	enough	in	nonetheless	something	twenty	whose
around	etc	inc	noone	sometime	two	why
as	even	indeed	nor	sometimes	under	will
at	ever	instead	not	somewhere	unless	with
be	every	into	nothing	still	unlike	within
became	everyone	is	now	stop	unlikely	without
because	everything	isn't	nowhere	such	until	won't
become	everywhere	it	of	taking	up	would
becomes	except	it's	off	ten	upon	wouldn't
becoming	few	its	often	than	us	yeah

been	fifty	itself	on	that	used	yes
before	first	last	once	that'll	using	yet
beforehand	five	later	one	that's	very	you
begin	for	latter	one's	that've	via	you'd
beginning	former	latterly	only	the	ve	you'll
behind	formerly	least	onto	their	was	you're
being	forty	less	or	them	wasn't	you've
below	found	let	other	themselves	we	your
beside	four	let's	others	then	we'd	yours
besides	from	like	otherwise	thence	we'll	yourself
between	further	likely	our	there	we're	yourselves
beyond	had	ltd	ours	there'd	we've	
billion	has	made	ourselves	there'll	well	

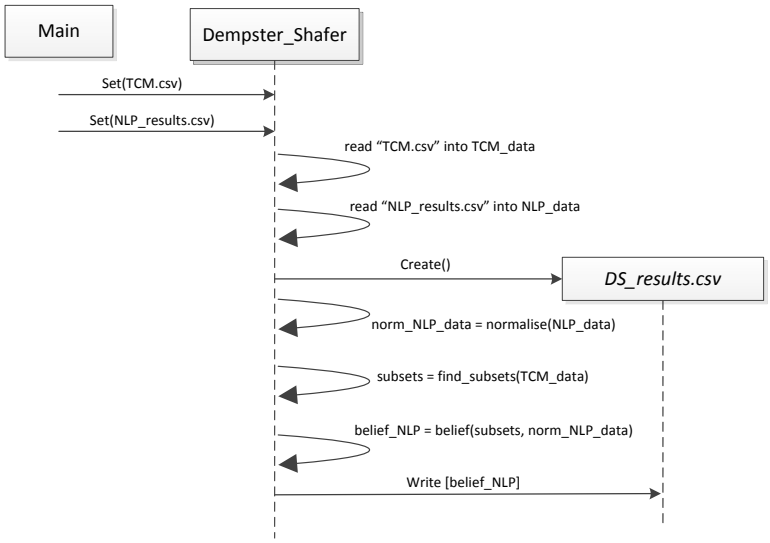
Appendix 3 – UML Diagrams



Grounded Theory Module



Dempster-Shafer Module



Appendix 4 – Python code

process_Google_results.py

```
path = '/home/natalia/Dropbox/newcode/MyCopy/links_cocaine/raw_it4_cocaine_cartels.txt'
txt_file = open(path)
txt = txt_file.read()
txt_file.close()
s_txt = txt.split(' ')
href_txt = [l for l in s_txt if l.startswith('href=')]
href_txt_1 = [l[6:-1] for l in href_txt if 'Webcache.google' not in l]
#href_txt_2 = [l for l in href_txt_1 if 'Webcache.google' not in l]
results_list = []
for line in href_txt_1:
    if line.startswith('http://'):
        if ('.google.') not in line:
            results_list += [line+'\n']

results_file = open('/home/natalia/Dropbox/newcode/MyCopy/it4_cocaine_cartels.txt', 'a') #
in append mode

for res in results_list:
    results_file.write(res)

results_file.close()
```

Main_NLP_module.py

```
'''
CHECK BEFORE RUN:
- all links are from the correct derictories
- kb_link path
- wp_link path
- tcm_file_name creates the file with the right name
- results_file_name creates the file with the right name
'''

from __future__ import division
import nltk, re, pprint, numpy, time
import URLlib2
from URLlib2 import URLopen
from nltk.corpus import stopwords
from collections import defaultdict
from functools import partial
from numpy import *
import phd_roget_open, phd_open_text_file, phd_NLP
import AlchemyAPI

import csv

start_time = time.time()


'''
############## DS Currency (Time Stamp) evaluation ############
>>> URL = "http://www.city.ac.uk"
>>> req = URLlib2.Request(URL)
>>> URL_handle = URLlib2.URLopen(req)
>>> headers = URL_handle.info()
>>> etag = headers.getheader("ETag")
>>> last_modified = headers.getheader("Last-Modified")
>>> last_modified
'''

# GET STOPWORDS
sw_link = '/home/natalia/Dropbox/newcode/MyCopy/stopwords.txt'  # stopwords file path
sw_text = phd_open_text_file.extract_text(sw_link)  # get stopwords tokens
stopwords = set([words for words in sw_text])       # get final list of stopwords

#####################################################################
```

```
concept_number = 1044         # concepts in Roget's thesaurus
#########################################################################
"""
path = ['C:/Python27/Lib/site-packages/heads/head'] # Roget's thesaurus file path
heads = phd_roget_open.roget_open(path, concept_number, stopwords)     # open thesaurus
txt file and transform into list
                                # [['word', 'index1', 'index2', ..., 'indexN'], ['word', 'index1', '..'], ...]
"""
#Path to Roget's Thesaurus txt file
path = '/home/natalia/Dropbox/newcode/MyCopy/roget_copy.txt'
heads = phd_roget_open.roget_open(path)

# GET KNOWLEDGE BASE
kb_link = '/home/natalia/Dropbox/newcode/MyCopy/KB_cocaine_production.txt'  # initial
knowledge base file path
kb_text = phd_open_text_file.extract_text(kb_link)          # get KB text tokens
kb_words = tuple([w.lower() for w in kb_text if w.isalpha()])      # all in lower case, get rid of
digits and spec symbols
kb_content = tuple([w for w in kb_words if w not in stopwords and len(w)>2])   # add to
content if not in stopwords and >2 letters

zero_idf = numpy.zeros(concept_number)  # inverted document frequency of concepts
(categories)

#-------------------------------------------------------------------------------
# PREPAPRE KB content for further processing (outside the loop)
#-------------------------------------------------------------------------------

class Text_Params:
    def __init__(self, content):
        self.content = content
        self.sorted_content = sorted(set(self.content)) # unique KB words in alphabetical order
        self.words_number=len(self.content) #  ... for both texts
        self.text_concept_matrix = phd_NLP.build_word_concept_matrix(self.sorted_content,
concept_number, heads)
        self.dup_matrix = phd_NLP.find_duplicates(self.content, self.sorted_content)
        self.WCCM = phd_NLP.build_WCCM(self.dup_matrix, self.text_concept_matrix,
self.words_number, concept_number)
        self.unique_words = len(self.sorted_content)
        self.merge_WCCM = phd_NLP.merge_WCCM_values(self.content, self.sorted_content,
concept_number, self.WCCM)
        self.SOA = phd_NLP.SOA_matrix(self.unique_words, concept_number,
self.merge_WCCM)
        self.TCM = self.text_concept_matrix.T  # ... transpose for further processing
```

```
        self.norm_vector = zero_idf  # for detecting which concepts are present in the text

class Measure_Similarity:
    def __init__(self, DP_wp, DP_kb):
        self.DP_wp = DP_wp
        self.DP_kb = DP_kb
    def cosDP(self):
        '''matrix of Cosines between distrib profiles of concepts'''
        self.cosDP = numpy.array(numpy.zeros((concept_number, concept_number))) # matrix
of Cosines between distrib profiles of concepts

        self.DP_wp_T = self.DP_wp.T # Transpose both DP matrices for further processing
        self.DP_kb_T = self.DP_kb.T

        _wp_concept_index = -1 # -1 to start with the first vector
        for wp in self.DP_wp_T:    # for every row (vector) in Web-page content distrib profile
matrix
            _wp_concept_index += 1 # add 1 -> start with the first row
            _kb_concept_index = -1 # same, -1 to start with the first vector
            for kb in self.DP_kb_T:    # for every row (vector) in KB distrib profile matrix
                _kb_concept_index += 1 # same, # add 1 -> start with the first row
                self.cosDP[_wp_concept_index,_kb_concept_index] =
phd_NLP.cosine_distance(wp,kb) #fill matrix with Cos distances between concept vectors
        self.cosDP[numpy.isnan(self.cosDP)] = 0  # replace Not-A-Number (NaN) values with
zeros
        return self.cosDP


def pre_process_wp_text(wp_text):
    '''cleans Web-text from stopwords and AlchemyAPI prefix'''
    raw = nltk.clean_html(wp_text)
    tokens = nltk.word_tokenize(raw)
    wp_text = nltk.Text(tokens)
    wp_text = wp_text[30:]
    wp_words = [w.lower() for w in wp_text if w.isalpha()]
    wp_content = [w for w in wp_words if w not in stopwords and len(w)>2]
    return wp_content


def fill_tf_vector(merge_WCCM):
    '''creates a vector with TF values for each of 1044 concepts'''
    merge_WCCM_T = merge_WCCM.T # transpose to be able to add all values for each
concept by row
    tf_vector = []
```

```
    for i in xrange(0,concept_number):
        concept_sum = sum(merge_WCCM_T[i])
        tf_vector.append(concept_sum)
    return tf_vector

def text_similarity(tf_vector, cosDP):
    '''calculates similarity between two texts, uses TF for weighting concepts'''
    left_num = 0         # left numerator - Web-page text concepts to KB
    right_num = 0        # right numerator - KB concepts to Web-page text
    # calculating left part of equation (candidate text concepts to KB)
    i=-1
    tf_sum = 0           #denominator
    for v in cosDP:
        i += 1
        left_num += v.max() * tf_vector[i]  # numerator
        tf_sum += tf_vector[i]        # denominator
    wp_to_kb = left_num/tf_sum  # left part of equation: Web-page similarity with respect to
KB

    cosDP_T = cosDP.T  # transpose, DP vectors for concepts in KB with resp. to page text
    # calculating right part of equation (KB concepts to candidate text)
    i=-1     # -1 to start with the first vector
    tf_sum = 0
    for v in cosDP_T:
        i += 1    # start with the first vector
        right_num += v.max() * tf_vector[i] # numerator
        tf_sum += tf_vector[i]            # denominator
    kb_to_wp = right_num/tf_sum  # right part of equation: KB similarity with respect to
Web-page
    text_sim = (wp_to_kb + kb_to_wp)/2  # TOTAL SIMILARITY between two texts
    return text_sim

KB = Text_Params(kb_content) # Knowledge Base
KK = KB.content # Known Knowns - for GT part
#---------------------------------------------------------------------------

# Extract Web-page content

########  CREATE ALCHEMY_API OBJECT + LOAD KEY  #########
# Create an AlchemyAPI object.                        #
alchemyObj = AlchemyAPI.AlchemyAPI()                  #
                                                      #
# Load the API key from disk.                         #
alchemyObj.loadAPIKey("api_key.txt")                  #
```

```
#############################################################


#==== CHANGE THE LIST OF LINKS HERE
==================================
#wp_link =
"/home/natalia/Dropbox/newcode/MyCopy/it1_list_cocaine_production_and_distribution.txt"
#wp_link = "/home/natalia/Dropbox/newcode/MyCopy/it2_list_cocaine_trafficking_uk.txt"
#wp_link = "/home/natalia/Dropbox/newcode/MyCopy/it3_list_cocaine_smuggling_uk.txt"
wp_link = "/home/natalia/Dropbox/newcode/MyCopy/it4_list_cocaine_cartels.txt"
#=============================================================
=====

lines = tuple(line.strip() for line in open(wp_link))

results = []

##############################
tcm_file_name = "cocaine_sameKB_TCM_test_it4.csv"  # Pre DS csv file with 1/0 values
for concepts usage
fl = open(tcm_file_name, 'wb')
TCM_file = csv.writer(fl)
##############################


##############################
results_file_name = "cocaine_sameKB_results_it4.csv"  # Pre DS csv file with 1/0 values for
concepts usage
res_fl = open(results_file_name, 'wb')
results_file = csv.writer(res_fl)
##############################

for line in lines:
    try:
        wp_text = alchemyObj.URLGetText(line)
    except:
        try:
            wp_text =alchemyObj.HTMLGetText(line, "http://www.test.com/")
        except:
            wp_text = 'Could not open URL'
            results = [line, '0', '0']
            results_file.writerow(results)
    #results += [wp_text]
    if wp_text != 'Could not open URL':
```

```
    wp_text = wp_text[340:]
    if len(wp_text)!=0:
        wp_content = pre_process_wp_text(wp_text)
        WP = Text_Params(wp_content)
        if len(WP.text_concept_matrix) != 0:
        #if WP.TCM!=[]: # avoid zero-size array, i.e. unknown words to thesaurus

            ####### PRE DS - CREATE SCV WITH 1/0 VALUES ##################
            ######  (1 - CONCEPT USED, 0 - CONCEPT NOT USED) #############
            tcm = WP.text_concept_matrix
            summ = numpy.array(tcm[0])
            for tcm_line in tcm[1:]:
                tcm_line = numpy.array(tcm_line)
                summ += tcm_line
            sum_i = -1
            for value in summ:
                sum_i += 1
                if value !=0:
                    summ[sum_i] = 1
            TCM_file.writerow(summ)


            ############ NLP ANALYSIS ####################################
            ##############################################################
            #_____________________________________________________________
            # Create Distributional Profiles of concepts
            merge_words = []   # augment two lists of unique words
            for row in WP.sorted_content:
                merge_words += [row]
            for row in KB.sorted_content:
                merge_words += [row]

            # set DP in alphabetical order, exclude duplications
            # len_DP - total amount of unique words for both texts
            # equals to the length of list of distributional profiles
            merge_words = sorted(set(merge_words))
            len_DP = len(sorted(set(merge_words)))

            # initialise matrices for Distributional Profiles of Concepts for both texts
            DP_wp = phd_NLP.create_DP(len_DP, concept_number, merge_words,
WP.sorted_content, WP.SOA)
            DP_kb = phd_NLP.create_DP(len_DP, concept_number, merge_words,
KB.sorted_content, KB.SOA)
```

```
        #Measure similarity between concepts
        cos_matrix = Measure_Similarity(DP_wp, DP_kb)
        cosDP = cos_matrix.cosDP() #build matrix of distrib profiles of concepts

        # Fill IDF values for concepts in idf matrix
        # idf = fill_idf_matrix(WP.TCM, KB.TCM)
        KB_tf_vector = array(fill_tf_vector(KB.merge_WCCM))
        WP_tf_vector = array(fill_tf_vector(WP.merge_WCCM))
        tf_vector = KB_tf_vector + WP_tf_vector

        #Calculate text similarity between KB and WP
        text_sim = text_similarity(tf_vector, cosDP)


        #~~~~~~~~ Texts are ADDED to the KB~~~~~~~~~~~~~~~

        #if text_sim > 0.85:              # if semantically close, add text to the Knowledge
Base
        #    kb_content += tuple(WP.content) # ...so that the next text would be compared
to the extended KB
        #    KB.content += tuple(WP.content)
        #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        # ~~~~~~~~Text are NOT ADDED to the KB ~~~~~~~~~~~~~~~~~~
        if text_sim > 0.85:
            KU_new = set([words for words in WP.sorted_content if words not in KK])
            KU_amount = len(KU_new)
            #KK += tuple(WP.content)
            ###kb_content += tuple(WP.content)
            #KB.content += tuple(WP.content)
            #KK = KB.content

        else:
            KU_amount = 0


        ############## GROUNDED THEORY ANALYSIS ###################
        #KU_new = set([words for words in WP.sorted_content if words not in KK])
        #KU_amount = len(KU_new)
        #KK += tuple(KU_new) # see KK rewrittten below KK = KB.content


        #str_KU_amount = str(KU_amount)
        ############################################################
```

```
            #print (line + ' ' + str(text_sim) + ' ' + str(KU_amount) + '\n')
            #run_time = (time.time() - start_time)
            #print (run_time)
            #results_file.write(line + ' ' + str(text_sim) + ' ' + str(KU_amount) + '\n')


            #KB = Text_Params(kb_content) # Rewrite Knowledge Base
            #KK = KB.content # Known Knowns - for GT part


            results = [line, text_sim, KU_amount]
            results_file.writerow(results)

fl.close() # PRE DS csv file close
res_fl.close() # close results file

#results_file.close()

run_time = (time.time() - start_time)
print (run_time)
```

Grounded_Theory.py

```
'''
Input links file: COPY two columns (links and Closeness) from the results+graphs file
  in a txt file (two columns will be tab delimited)

CHECK input KB file path + output csv file name

'''

import AlchemyAPI, nltk, numpy, csv
import URLlib2
from URLlib2 import URLopen
import phd_open_text_file
from nltk.corpus import stopwords

################################################################################
#
# Link to KB
kb_link = '/home/natalia/Dropbox/newcode/MyCopy/KB_cocaine_production.txt'  # initial
knowledge base file path
#
# path to txt doc with links+NLP
path =
"/home/natalia/Dropbox/newcode/MyCopy/PHD_results/Same_KB_cocaine/GT/cocaine_lin
ks_NLP.txt"

# Where the results will be stored
GT_results_file_name = "GT_cocaine_SameKB_results.csv"  # GT csv file with new KK
amount
#
# Where the unique words will be stored
GT_unique_words =
open('/home/natalia/Dropbox/newcode/MyCopy/GT_cocaine_SameKB_unique_words.txt',
'a')
#
################################################################################


# GET STOPWORDS
sw_link = '/home/natalia/Dropbox/newcode/MyCopy/stopwords.txt'  # stopwords file path
sw_text = phd_open_text_file.extract_text(sw_link)  # get stopwords tokens
stopwords = set([words for words in sw_text])           # get final list of stopwords
```

```
# (- !!! -   CHECK   - !!! -)  GET KNOWLEDGE BASE
#kb_link = '/home/natalia/Dropbox/newcode/MyCopy/KB_tobacco_industry.txt'  # initial knowledge base file path
kb_text = phd_open_text_file.extract_text(kb_link)          # get KB text tokens
kb_words = [w.lower() for w in kb_text if w.isalpha()]      # all in lower case, get rid of digits and spec symbols
kb_content = [w for w in kb_words if w not in stopwords and len(w)>2]   # add to content if not in stopwords and >2 letters


class Text_Params:
    def __init__(self, content):
        self.content = content
        self.sorted_content = sorted(set(content)) # unique KB words in alphabetical order
        #self.words_number=len(self.content) #  ... for both texts
        #self.unique_words = len(self.sorted_content)

def pre_process_wp_text(wp_text):
    '''cleans Web-text from stopwords and AlchemyAPI prefix'''
    raw = nltk.clean_html(wp_text)
    tokens = nltk.word_tokenize(raw)
    wp_text = nltk.Text(tokens)
    wp_text = wp_text[30:]
    wp_words = [w.lower() for w in wp_text if w.isalpha()]
    wp_content = [w for w in wp_words if w not in stopwords and len(w)>2]
    return wp_content


#find those with closeness > 0.85 and calculate KU amount
################################################################################

KB = Text_Params(kb_content) # Knowledge Base

#----- SWITCH THAT FOR THE FIRST USE ----------------
#-----------------------------------------------------
KK = KB.content # Known Knowns - for GT part
#KK_txt = open('/home/natalia/Dropbox/newcode/MyCopy/cocaine_GT_unique_words.txt', 'r')
#raw_KK = KK_txt.read()
#KK = raw_KK.split()
#KK_txt.close()

#-----------------------------------------------------------------------
```

```
# Extract Web-page content

########  CREATE ALCHEMY_API OBJECT + LOAD KEY  #########
# Create an AlchemyAPI object.                          #
alchemyObj = AlchemyAPI.AlchemyAPI()                    #
                                                        #
# Load the API key from disk.                           #
alchemyObj.loadAPIKey("api_key.txt")                    #
############################################################

#read txt file with ALL(!!!) links and corresponding NLP results
#path =
"/home/natalia/Dropbox/newcode/MyCopy/PHD_results/Cumulative_KB_results_tobacco_to
bacco/GT/tobacco_AddKB_links_NLP.txt"
fl = open(path)
lines = fl.read()
split_lines = lines.splitlines()
my_lines = []
links = []
NLP = []
for line in split_lines: #         !!!!!       R E M O V E [:20]
   link,closeness = line.strip().split("\t")
   links.append(link)    # list of Google results (links)
   NLP.append(closeness) # corresponding list of NLP values
fl.close()
# there are now two lists: links and corresponding NLP results

#lines = [line.strip() for line in open(wp_link)]

results = []

##############################
#GT_results_file_name = "GT_tobacco_AddKB_TT_results.csv"  # GT csv file with new
KK amount
GT_res_fl = open(GT_results_file_name, 'a')
GT_results_file = csv.writer(GT_res_fl)
##############################

#this is where we're going to add all unique words throughout iterations
#GT_unique_words =
open('/home/natalia/Dropbox/newcode/MyCopy/GT_tobacco_AddKB_TT_unique_words.txt'
, 'a')
```

```
cum_KU = 0
'''
for link in links[:5]:
    try:
        wp_text = alchemyObj.URLGetText(line)
    except:
        try:
            wp_text =alchemyObj.HTMLGetText(line, "http://www.test.com/")
        except:
            wp_text = 'Could not open URL'

    if wp_text != 'Could not open URL':
        wp_text = wp_text[340:]
        if len(wp_text)!=0:
            wp_content = pre_process_wp_text(wp_text)
            WP = Text_Params(wp_content)
            if len(WP.content) != 0:
                ############## GROUNDED THEORY ANALYSIS ###################
                #KK = KB.content
                KU_new = set([words for words in WP.sorted_content if words not in KK])
                KU_amount = len(KU_new) # amount of new words in this text
                cum_KU += KU_amount  # Calculate cumulative summ of KU
                for KU in KU_new:
                    KK += [KU]       # Add new words to the list of unique words for all the texts
                GT_results = [str(link), KU_amount, cum_KU]
                GT_results_file.writerow(GT_results)
                ###########################################################

'''
NLP_ind = -1 # index of NLP_result and corresponding WP link
for NLP_result in NLP: # calculate new KU for all links texts
    NLP_ind +=1
    if float(NLP_result) > 0.85: # consider ALL links with semantic closeness > 85%
        line = links[NLP_ind]
        try:
            wp_text = alchemyObj.URLGetText(line)
        except:
            try:
                wp_text =alchemyObj.HTMLGetText(line, "http://www.test.com/")
            except:
                wp_text = 'Could not open URL'
                #results = [links[NLP_ind], '0']
                #GT_results_file.writerow(results)
        #results += [wp_text]
```

```
        if wp_text != 'Could not open URL':
            wp_text = wp_text[340:]
            if len(wp_text)!=0:
                wp_content = pre_process_wp_text(wp_text)
                WP = Text_Params(wp_content)
                if len(WP.content) != 0:
                    ############## GROUNDED THEORY ANALYSIS ###################
                    #KK = KB.content
                    KU_new = set([words for words in WP.sorted_content if words not in KK])
                    KU_amount = len(KU_new) # amount of new words in this text
                    cum_KU += KU_amount  # Calculate cumulative summ of KU
                    for KU in KU_new:
                        KK += [KU]       # Add new words to the list of unique words for all the texts
                    GT_results = [line, KU_amount, cum_KU]
                    GT_results_file.writerow(GT_results)
                    ###########################################################
                    #print line, NLP_result, KU_amount

for item in KK:
    GT_unique_words.write(item+' ')

GT_unique_words.close()

GT_res_fl.close()
```

Dempster_Shafer_Module.py

```
'''
DS module. Calculates Belief and Plausibility. Requires csv files:
1. Matrix of True/False values for concept usage (columns) in the documents (rows)
2. Matrix of mass functions(rows) for each source of evidence(columns)
'''

import csv
import numpy

def csv_to_array(csv_file_path):
    '''reads from csv into array of rows'''
    csv_file = open(csv_file_path, 'rb')
    data = csv.reader(csv_file, delimiter = ',')
    array = []
    ###################################
    for row in data:
        array += [row]
    array = tuple(array)
    csv_file.close()
    return array # [-1] when last row in TCM.csv is not complete

def check_subset(set_A, set_B):
    '''checks if set_B is a subset of set_A. Sets are 1044 each with True values corresponding
to the concepts used'''
    result = True
    if len(set_A) != len(set_B):
        raise 'Sets must be of the same length!'
    for i in xrange(0,len(set_A)):
        if set_B[i] == '1':
            if set_A[i] == '0':
                result = False
                break
    return result

def find_subsets(array):
    '''creates a list of [row index, ind, ind...] returns groups of subsets of every row within the
same array of rows'''
    array_copy = array # create copy of the array for iteration
    r = -1 # index will start from 0
    subsets = []
    for arr in array:
        r += 1
```

```
        group = [r] # row index - 1st element in a group of values [row index, ind,ind...]
        rr = -1 # index will start from 0
        for arr1 in array_copy:
            rr += 1
            if r != rr: # do not compare same rows
                if check_subset(arr, arr1) == True:
                    group.append(rr)
        subsets += [group]
    return subsets

def normalise(array, column):
    '''normalisation of array values for a column (with given index)'''
    norm_data = []
    data = [el[column] for el in array] # take first elements (NLP results) as Source_1 evidence
    summ = sum(data)
    for i in data:
        j = i/summ
        norm_data += [j]
    return norm_data

def belief(subsets, mass_data):
    '''calculates DS Belief for each document for the mass function corresponding to source of
evidence - attr_data[column]'''
    belief = [] # for one source of evidence
    for subset in subsets: # calculate Bel for each doc in array of subsets (i.e.for each row)
        bel = 0
        for s in subset:
            bel += mass_data[s]#[column] # source of evidence
        belief += [bel]
    return belief


def find_intersections(array):
    '''finds indecex of intersecting BOOL rows '''
    intersection = []
    arr_i = -1 # row index
    #array = concept_data
    for arr in array:
        array_copy = list(array) # create a renewable copy to iterate in it
        arr_i += 1 # increase row index
        group = [arr_i] # for each row remember row index and add intersection rows to it later
        gr = [] #list of indexes of intersecting rows to be added to [group]
        el_i = -1 # element index
        for el in arr: # iterate over row elements
```

```
            el_i +=1  # increase element index
            if el == 'TRUE':
                copy_i = -1 # same/copy_array row index
                for arr1 in array_copy:
                    copy_i += 1
                    if arr_i != copy_i: # do not compare row to itself
                        if arr1[el_i] == el:
                            gr.append(copy_i)
        group += set(gr)
        intersection += [group]
    return intersection


def plausibility(intersections, mass_data):
    '''calculates DS Plausibility for each document for the mass function corresponding to
source of evidence - attr_data[column]'''
    plausibility = [] # for one source of evidence
    for inter in intersections: # calculate Bel for each doc in array of subsets (i.e.for each row)
        pl = 0
        for s in inter:
            pl += mass_data[s]#[column] # source of evidence
        plausibility += [pl]
    return plausibility

################################################################################
####

path = 'iteration1_TCM_test.csv' # csv with True/False data on concepts used within the text
concept_data = csv_to_array(path) # read data to array (tuple)
subsets = find_subsets(concept_data)

path = 'iteration1_results_test_justnumbers.csv' # csv with NLP and other quality metrics of
the texts for DS
attr_data_str = csv_to_array(path) # read data to array (tuple)
attr_data = []
for a in attr_data_str[:265]: # convert all values to floats
    el = [float(i) for i in a]
    attr_data += [el]

norm_NLP_data = normalise(attr_data, 0)
belief_NLP = belief(subsets,norm_NLP_data)

intersections = find_intersections(concept_data)
plausibility_NLP = plausibility(intersections, norm_NLP_data)
```

```
#plausibility_NLP = plausibility_NLP_1[:265]

DS_results = numpy.array([belief_NLP]+[plausibility_NLP]).T

DS_file_name = "DS_results_test.csv"
fl = open(DS_file_name, 'wb')
DS_file = csv.writer(fl)
#for row in DS_results:
 #   DS_file.writerow(row)
DS_file.writerows(DS_results)
fl.close()
```

Appendix 5 – Model Testing Results

#	URL	*text_sim*	System Coding	Human Coding	Error
1	http://www.culturalsurvival.org/publications/cultural-survival-quarterly/bolivia/south-american-cocaine-production	0.977	Relevant	Relevant	
2	http://www.erowid.org/archive/rhodium/chemistry/cocaine.illicit.production.html	0.972	Relevant	Relevant	
3	http://www.umsl.edu/~rkeel/180/drug_industry.htm	0.973	Relevant	Relevant	
4	http://www1.american.edu/ted/ice/cocaine.htm	0.973	Relevant	Relevant	
5	http://www.justice.gov/archive/ndic/pubs07/796/cocaine.htm	0.973	Relevant	Relevant	
6	http://www.ncjrs.gov/App/publications/Abstract.aspx?id=169970	0.880	Relevant	Relevant	
7	http://www.justice.gov/archive/ndic/pubs1/1017/index.htm	0.956	Relevant	Relevant	
8	http://williamwarelaw.com/areas-of-practice/drug-crimes/cocaine-distribution-charges/	0.969	Relevant	Relevant	
9	http://www.newjerseycriminallawattorney.com/CM/DrugCrimes/PracticeAreaDescriptions53.asp	0.978	Relevant	Not Relevant	False Positive
10	http://evolutionofdruguse.wordpress.com/alcohol/cocaine/	0.971	Relevant	Relevant	
11	http://www.h-net.org/reviews/showrev.php?id=25701	0.977	Relevant	Relevant	
12	http://www.unodc.org/unodc/en/data-and-analysis/bulletin/bulletin_1950-01-01_4_page005.html	0.973	Relevant	Relevant	
13	http://www.drugs-forum.com/forum/showthread.php?t=139418	0.974	Relevant	Relevant	
14	http://www.inchem.org/documents/pims/plant/erythrox.htm	0.973	Relevant	Not Relevant	False Positive
15	http://www.npr.org/templates/story/story.php?storyId=126978142	0.971	Relevant	Not Relevant	False Positive
16	http://law.justia.com/codes/louisiana/2006/48/98881.html	0.922	Relevant	Relevant	
17	http://www.narconon.org/drug-information/methamphetamine-history.html	0.976	Relevant	Not Relevant	False Positive
18	http://www.tni.org/briefing/revolutionary-armed-forces-colombia-farc-and-illicit-drug-trade	0.972	Relevant	Relevant	
19	http://www.livesaymyers.com/criminal-lawyers/drug-charges/distribution/	0.974	Relevant	Not Relevant	False Positive

20	http://en.wikipedia.org/wiki/Domestic_policy_of_Evo_Morales	0.972	Relevant	Relevant	
21	http://www.vice.com/en_uk/vice-news/sisa-cocaine-of-the-poor-part-1	0.973	Relevant	Not Relevant	False Positive
22	http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1975811/	0.972	Relevant	Relevant	
23	http://www.drugfreeworld.org/drugfacts/crystalmeth/history-of-methamphetamine.html	0.968	Relevant	Not Relevant	False Positive
24	http://www.the-american-interest.com/article.cfm?piece=1316	0.972	Relevant	Not Relevant	False Positive
25	http://uatests.com/drug-information/cocaine.php	0.968	Relevant	Relevant	
26	http://ellblog.com/government-not-required-to-produce-esi-in-manner-requested-by-defendants-in-cocaine-distribution-conspiracy-case/	0.973	Relevant	Relevant	
27	http://english.nkradio.org/news/290	0.913	Relevant	Relevant	
28	http://www.mytripjournal.com/travel-542493-cocoa-museum-cocaine-production-wealthy-chew-corporations-distribution	0.957	Relevant	Relevant	
29	http://www.nytimes.com/1994/02/20/world/colombians-press-for-the-legalization-of-cocaine.html	0.960	Relevant	Relevant	
30	http://www.barnesandnoble.com/s/9781244794559	0.965	Relevant	Not Relevant	False Positive
31	http://aibrraiokd.blogdetik.com/cocaine-consumers-handbook-book-downloads/	0.972	Relevant	Relevant	
32	http://www.questia.com/library/1G1-135391435/failed-plan-after-five-years-and-billions-of-u-s	0.905	Relevant	Not Relevant	False Positive
33	http://correctionspageone.blogspot.com/2010/03/getting-even-on-crack-vs-powder.html	0.971	Relevant	Relevant	
34	http://www.mushroomgeeks.com/forum/showthread.php?t=26459	0.953	Relevant	Not Relevant	False Positive
35	http://www.hebert-law.com/index.php/prohibited-acts-schedule-ii-penalties-r-s-40-967.html	0.955	Relevant	Not Relevant	False Positive
36	http://vspages.com/cocaine-vs-caffeine-2-2900/	0.977	Relevant	Relevant	
37	http://bookverdict.com/details.xqy?uri=Product-9971189.xml	0.937	Relevant	Not Relevant	False Positive
38	http://www.fox47.com/newsroom/top_stories/videos/road-trippin-new-glarus-brewery-continues-its-climb-6256.shtml	0.978	Relevant	Not Relevant	False Positive
39	http://factsanddetails.com/world.php?itemid=1223&catid=54&subcatid=348	0.974	Relevant	Not Relevant	False Positive
40	http://uk.ask.com/beauty/Crack-Cocaine-Manufacture	0.955	Relevant	Relevant	
41	http://anxiety-pills.com/dea-drug-information-cocaine	0.971	Relevant	Relevant	

42	http://www.onlinedrugspills.com/cocaine-production-how-to	0.971	Relevant	Relevant	
43	http://www.experiencefestival.com/cocaine_-_africa	0.975	Relevant	Relevant	
44	http://en.goldenmap.com/Cocaine_Cowboys	0.976	Relevant	Relevant	
45	http://www.directessays.com/essays/Colombia/Cocaine.html	0.908	Relevant	Relevant	
46	http://www.uncorneredmarket.com/2009/12/cocaine-bolivian-jungle/	0.644	Not Relevant	Relevant	False Negative
47	http://www.ftlucianolaw.com/blog/marijuana-criminal-attorney-in-hackensack--hackensack----bolivia-cuts-coca-leaf-production.cfm	0.722	Not Relevant	Not Relevant	
48	http://www.pbs.org/thebotanyofdesire/altering-consciousness.php	0.642	Not Relevant	Not Relevant	
49	http://www.jstor.org/stable/2780533	0.761	Not Relevant	Not Relevant	
50	http://www.cablegatesearch.net/cable.php?id=07TORONTO90	0.793	Not Relevant	Not Relevant	
51	http://www.studyblue.com/notes/note/n/chapter-13/deck/7959	0.719	Not Relevant	Not Relevant	
52	http://www.ibtimes.com/	0.799	Not Relevant	Not Relevant	
53	http://news.silobreaker.com/fort-mcmurray-man-charged-after-cocaine-cash-seized-5_2266811707932278847	0.808	Not Relevant	Not Relevant	
54	http://www.mightystudents.com/tag/coca%20leaf%20cocaine%20production	0.795	Not Relevant	Relevant	False Negative
55	http://www.yatedo.com/s/movie%3A(Cocaine+Angel)	0.781	Not Relevant	Not Relevant	

Appendix 6 – Proof of Mathematics

The material in this Appendix aims to provide further mathematical background to the proposed semantic closeness measurement, described in detail in Section 4.3. As explained in Chapter 4 of this thesis, this research uses a hybrid method detailed in (Hirst & Mohammad, 2006) that combines the co-occurrence statistics with the information in a lexical source (this research uses Roget's Thesaurus).

The Distributional Hypothesis in linguistics is derived from the semantic theory of language usage, i.e. words that are used and occur in the same contexts tend to share similar meanings (Firth 1957; Harris 1968). When applied to word senses (instead of words), then the hypothesis states that words when used in different senses tend to keep different "company" (co-occurring words). Hirst & Mohammad (2006) proposed the creation of distributional profiles (DPs) of word senses or concepts, rather than those of words. The closer the distributional profiles of two concepts, the smaller is their semantic distance.

Determining distributional profiles of concepts requires information about which words co-occur with which concepts. Below is a way to estimate distributional profiles of concepts from raw text, using a published thesaurus, as described by Hirst & Mohammad.

First, word–category co-occurrence matrix (WCCM) is created having word types *w* as one dimension and thesaurus categories *c* as another:

	c_1	c_2	...	c_j	...
w_1	m_{11}	m_{12}	...	m_{1j}	...
w_2	m_{21}	m_{22}	...	m_{2j}	...
...	...	...	...	...	...
w_i	m_{i1}	m_{i2}	...	m_{ij}	...
...	...	...	...	...	...

The matrix is populated with co-occurrence counts from a large corpus. A particular cell m_{ij}, corresponding to word w_i and category or concept c_j , is populated with the number of times w_i co-occurs (in a window of ±5 words) with any word that has c_j as one of its senses (i.e., w_i co-occurs with any word listed under concept c_j in the thesaurus).

The choice of ±5 words as window size is somewhat arbitrary and hinges on the intuition that words close to a target word are more indicative of its semantic properties than those more distant. Church and Hanks (1990), in their seminal work on word–word co-occurrence association, also use a window size of ±5 words and argue that this size is large enough to capture many verb–argument dependencies and yet small enough that adjacency information is not diluted too much.

A contingency table for any particular word *w* and category *c* can be easily generated from the WCCM by collapsing cells for all other words and categories into one and summing up their frequencies:

	c	$\neg c$
w	n_{wc}	$n_{\neg w}$
$\neg w$	$n_{\neg c}$	$n_{\neg\neg}$

The application of a suitable statistic, such as pointwise mutual information or conditional probability, will then yield the strength of co-occurrence association between the word and the category. Below are example distributional profiles of two senses of STAR:

CELESTIAL BODY: space 0.36, light 0.27, constellation 0.11, hydrogen 0.07, . . .
CELEBRITY: famous 0.24, movie 0.14, rich 0.14, fan 0.10, . . .

Once created, any of the many measures of vector distance can be used to estimate the distance between the DPs of two target concepts.

Based on the detailed survey of semantic distance measures in (Hirst & Mohammad, 2006), the algorithm proposed in this thesis uses the adapted Cosine method to estimate distributional distance between two concepts. Cosine similarity is an accepted measure of similarity between two vectors that measures the Cosine of the angle between them. Given two vectors of attributes, *A* and *B*, the Cosine similarity *Cos*(*θ*) is calculated using a dot product and magnitude:

$$Cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}. \qquad \text{(A6-1)}$$

In essence, vectors *A* and *B* are Distributional Profiles (DPs). The attributes of each of the vector are strength of association (SOA) values as per the DP. For example, applying the above formula, here is how Cosine is traditionally used to estimate distributional distance between two words using their DPs:

$$Cos(\mathrm{w}_1, \mathrm{w}_2) = \frac{\sum_{w \in C(\mathrm{w}_1) \cup C(\mathrm{w}_2)} (P(w \mid \mathrm{w}_1) \times P(w \mid \mathrm{w}_2))}{\sqrt{\sum_{w \in C(\mathrm{w}_1)} (P(w \mid \mathrm{w}_1)^2} \times \sqrt{\sum_{w \in C(\mathrm{w}_2)} (P(w \mid \mathrm{w}_2)^2}}, \qquad \text{(A6-2)}$$

C(w) is the set of words that co-occur (within a certain text window) with the word *w* in a corpus. The conditional probabilities in the formula are taken from the distributional profiles of words. Replacing word co-occurrence with word-concept co-occurrence in (A6-2), the Cosine distributional distance measure for two concepts c_1 and c_2 is, therefore, denoted by:

$$Cos(c_1, c_2) = \frac{\sum_{w \in C(c_1) \cup C(c_2)} (P(w \mid c_1) \times P(w \mid c_2))}{\sqrt{\sum_{w \in C(c_1)} (P(w \mid c_1)^2} \times \sqrt{\sum_{w \in C(c_2)} (P(w \mid c_2)^2}}, \qquad \text{(A6-3)}$$

C(x) is now the set of words that co-occur with concept *x* within a pre-determined text window. The conditional probabilities in the formula are taken from the distributional profiles of concepts. If the distance between two words is required, and their intended senses are not known, then the distance between all relevant sense pairs is determined and the minimum is chosen.

The algorithm proposed in this thesis creates a matrix populated with Cosine values for each pair of Thesaurus concepts between two texts:

	$c_1(T_2)$	$c_2(T_2)$	...	$c_{1044}(T_2)$
$c_1(T_1)$	$Cos(c_1(T_1), c_1(T_2))$	$Cos(c_1(T_1), c_2(T_2))$	...	$Cos(c_1(T_1), c_{1044}(T_2))$
$c_2(T_1)$	$Cos(c_2(T_1), c_1(T_2))$	...	...	...
...	...	...	...	...
$c_{1044}(T_1)$	$Cos(c_{1044}(T_1), c_1(T_2))$	....	...	$Cos(c_{1044}(T_1), c_{1044}(T_2))$

As explained in Chapter 4 of this thesis, such Cosine values are then used as input to the formula by Corley & Mihalcea (2005), which has been specifically adapted for this research. The original authors' approach attempts to model the semantic similarity of texts as a function of the semantic similarity of the component words by combining metrics of word-to-word similarity and language models into a formula that is a potentially good indicator of the semantic similarity of the two input texts, when based on distributional information learned from large corpora.

While word frequency does not always constitute a good measure of word importance, the distribution of words across an entire collection can be a good indicator of the specificity of the words. Terms that occur in a few documents with high frequency contain a greater amount of discriminatory ability, while terms that occur in numerous documents across a collection with a high frequency have inherently less meaning to a document. Following on that, Corley & Mihalcea (2005) determine the specificity of a word using the inverse document frequency which is defined as:

$$idf(w) = \frac{N_{total}}{N_w}, \tag{A6-4}$$

where N_{total} is the total number of documents in the corpus, and N_w is the total number of documents that include the word w. Provided a measure of semantic similarity between words, and an indication of the word specificity $idf(w)$, they can be combined into a measure of text semantic similarity, by pairing up those words that are found to be most similar to each other, and weighting their similarity with the corresponding specificity score.

In the original method, for a given pair of texts, sets of nouns, verbs, adjectives, and adverbs are defined in one text with determined pairs of similar words sets across the other text. For example, for each noun belonging to one of the text segments, the noun in the other text segment is identified that has the highest semantic similarity (*maxSim*). The similarity between the input text segments T_i and T_j is then determined using a scoring function that combines the word-to-word similarities and the word specificity:

$$sim(T_i, T_j)_{T_i} = \frac{\sum_{type}\left(\sum_{w_k \in \{W_{type}\}} (\max Sim(w_k) * idf(w_k))\right)}{\sum_{w_k \in \{Ti\}} idf(w_k)}. \tag{A6-5}$$

This score, which has a value between 0 and 1, is a measure of the directional similarity, in this case computed with respect to T_i. The scores from both directions can be then combined into a bidirectional similarity using an average function:

$$sim(T_i, T_j) = \frac{1}{2}\left(sim(T_i, T_j)_{T_i} + sim(T_i, T_j)_{T_j} \right). \quad \text{(A6-6)}$$

Specifically for this research, the formula (A6-5) has been adjusted replacing word-to-word similarity with concept-to-concept similarity, previously obtained using distributional profiles of concepts, in order to improve semantic properties of the search. From the matrix of Cosine values (see above), for each concept $c_i(T_1)$ in the text T_1 the concept $c_j(T_2)$ in the text T_2 is identified that has the highest semantic similarity Cos($c_i(T_1)$, $c_j(T_2)$)), which will be denoted as max $Sim(c, T_2)$. Similarly, for each concept $c_j(T_2)$ in the text T_2, the concept with the highest semantic similarity is identified in the text T_1. This allows calculating a metric that combines semantic similarity of concepts in text T_1 with respect to concepts in text T_2. Substituting words references with concepts in (A6-5) and applying this to the formula (A6-6), semantic similarity score becomes:

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{c \in \{T_1\}} (\max Sim(c, T_2) * idf(c))}{\sum_{c \in \{T_1\}} idf(c)} + \frac{\sum_{c \in \{T_2\}} (\max Sim(c, T_1) * idf(c))}{\sum_{c \in \{T_2\}} idf(c)} \right). \quad \text{(A6-7)}$$