Provision of better VLE learner support with a Question Answering System

Marios Nikolaos Stamoulos

A thesis/doctoral report and portfolio submitted in partial fulfilment of the for the degree of Doctor of Philosophy requirements of the University of Sunderland PhD by Existing Published or Creative Works.

August 2016

Abstract

The focus of this research is based on the provision of user support to students using electronic means of communication to aid their learning. Digital age brought anytime anywhere access of learning resources to students. Most academic institutions and also companies use Virtual Learning Environments to provide their learners with learning material. All learners using the VLE have access to the same material and help despite their existing knowledge and interests.

This work uses the information in the learning materials of Virtual Learning Environments to answer questions and provide student help by a Question Answering System.

The aim of this investigation is to research if a satisfactory combination of Question Answering, Information Retrieval and Automatic Summarisation techniques within a VLE will help/support the student better than existing systems (full text search engines).

Acknowledgements

I would like to thank my supervisors Dr. Chris Bowerman and Dr. Michael Oakes for their continuous support, advice and patience during my research. Without their support it would not be possible to complete my research. They gave me their unlimited attention and kept me in the right path in order to complete my degree.

I would also like to thank all the staff from the University of Sunderland that in multiple instances they helped in in completing my degree and overcome situations. It has been a pleasure working with such a great professionals.

I would like to say a huge thank you to my family that supported me through the years in order to come to the end of this work. They stood next to me on the late nights, encouraged every step I made and supported me above and beyond any expectation. My wife, has been encouraging me every time I thought I hit a wall and helping me to get over any of the problems. My kids were giving me motivation to work hard in order to be a good role model. My mother and auntie, although being away, they were just a phone call away in order to help with anything they could.

Finally, I could not exclude from the acknowledgments some people that supported me and guided me in my earlier years and are not with me at this time. Mr. G. Bookis, that has helped me during my earlier education which provided me with strong foundations for my studies. My grandmother and godmother that established timeless principles of working hard to achieve your dreams, that I carry throughout my life.

Contents

Table of figures

Table of tables

Chapter 1

1 Introduction

1.1 Motivation of research

This describes about research undertaken in order to investigate if a statistical based Question Answering system can provide accurate responses and enhance the student experience within a Virtual Learning Environment. E-Learning is a standard method used to deliver material to the user. The way the internet technologies work now is completely different from how it was 15 years ago. There was a phase where highly customised projects were developed to serve specific learning tasks (Willliams, Goldberg, 2005). This proved not fit for purpose, since the tools were very expensive and were not widely used.

In the last decade the internet has changed from relatively static pages to dynamic web apps that integrate with other systems, serve low latency multimedia content to a global user base with various browsers often on mobile devices rather that desktop computers (KendeI, 2012).

In the e-learning area, there is also a great progress in the amount of content available, where more and more content is stored online. A considerable change has been made in the way this content is delivered to students and also assessing their knowledge. Content reuse, although possible though the online repositories that the content is stored in, is not often achieved. As part of this thesis, we will take advantage of the content available and use statistical approaches in order to support the students through their learning experience and enable content reuse.

As more learning content is being stored online and is accessible to the student, there is a clear need of support while the students read the

materials online. Usually internet search engines come to the help of the student, but there is a great chance that the help content that the student needs to work on would be missed. Also getting support from academic staff, is difficult considering the amount of classes and the times the students will be working on their assessments or study. Forums are used to support the students may provide certain full text search features, but the information stored in the forums may be out of date, incorrect and also limited.

What we seek to investigate in this research, is the provision of support to the learner using existing learning materials to which the student has access. We want the algorithms used by the system to depend only on statistical methods. The reason behind that is that we don't want any involvement of the academic staff in order to support the students via our proposed automatic solution.

What we would like to do is to make use of a question answering system in the e-learning domain using a more generic approach that requires the least human intervention for the system to provide the correct answer to the student. To our knowledge, the ones that have been implemented require an expert user's intervention, which would be the teaching staff. Although this would be acceptable for a bespoke system, it adds an extra layer of work on the already overloaded teaching staff. Requiring a domain expert to support and maintain the application is not something that could be done in a widely used system.

Most VLEs have some functionality that allows full text searching of the documents uploaded. The level of support a search engine can provide is limited, especially when the data stored in the system is large. Also the results of the search engine will provide a list of related documents to the query which the user then needs to look into. The difference between a search engine and a question answering system is that in the latter the answer the user is looking for would be pinpointed. This will help the user in easily retrieving context extracts of interest in response to a query and will help the learner study online without the need to search for documents while working on projects or assignments.

All the above were the main reasons for starting this project. There is a clear gap between the tools for authoring and delivering content and the tools that assess students. This gap includes support tools for when the student studies online.

In the next section, we will introduce the definition of Question Answering systems to the reader in order to make clear how such system can provide support within a Learning Environment.

1.2 Definition of QA

Question answering (QA) systems are systems that return a single answer to a user's query rather than a single or a set of documents or the document that contains the information used to compile the answer (Voorhees, 2004). Such technologies are not present in Virtual Learning environments. Research on QA is not something that has started in the last few years. There are two main categories of research in this field: approaches that

depend on lexical, syntactic and other knowledge bases, and approaches that exploit the statistics of the corpus in order to provide an answer. Of course these categories are not clearly separated and there are systems developed that use both approaches in their implementation. In all of the systems we can see that there are three clearly identified processing stages that perform different tasks. These are: the Question Parsing functionality, where user's question is passed through a series of algorithms in order to extract the maximum amount of information from the question's text, the Document Retrieval functionality, where a set of algorithms is used in order to identify documents that contain information related to the question and finally the Answer Pinpointing functionality where the most relevant answer is extracted or generated from the set of documents identified at the previous stage and is then presented back to the user as the answer.

1.3 Aims and objectives

The aim of this investigation is to identify and develop a satisfactory combination of Question Answering and Information Retrieval techniques within a VLE to support the student better than existing systems (full text search engines). The features that will be used in order to evaluate our system against the baseline would be:

- Using Cisco CCNA self-assessment questions as input, the system should answer correctly using a passage from the CCNA corpus. The selected passage should be part of the top selected document if the same question is fed into the baseline system (search engine).

- Our system should provide the answer with less clicks than the baseline system
- Using our system, the students should be able to get the correct answer in less time, since they will not have to look into multiple documents and also by looking in less documents.
- The information provided to the user should contain less irrelevant data than the baseline system
- The users should prefer to use the QA system instead of the search engine

The questions we use are taken from the self-assessment quizzes at the end of each chapter in the CCNA online notes. The answers will also be picked form the Cisco CCNA online notes. Cisco's CCNA notes are used widely by networking practitioners all over the world as Cisco CCNA is one of the main networking qualifications.

1.4 Research question and hypothesis

In this research work, we use statistical methods in order to support students within a VLE by answering their questions while they study online materials. Having identified the main issues in Question Answering systems and realising that in order for the system to support the maximum amount of students the system should operate without any input from the teaching staff, the research questions that arise are:

RQ1. Can a QA system using statistical based techniques provide the similar level of answers as a baseline search engine?

RQ2. How well will our algorithm work using a smaller or a larger corpus since the amount of documents in the VLE will be different per institution.

RQ3. Is there a categorisation technology such as topic signatures that can be improved from the current state and used within our algorithm in order to support students.

RQ4. Summarisation is an Information Retrieval technique that, given a document, returns the important sentences of a document. Would that kind of technique be of use for choosing the answer to a user question?

From the questions above we derive the following hypotheses:

H1. The correct answer to a question entered to the system should be retrieved using statistical methods and without requiring any background knowledge.

H2. The statistical approaches used will not be dependent on the size of the corpus and the system should be able to retrieve the correct answer having a small or a large corpus to use for weight calculations.

H3. A good combination of methods that will work on a learning domain to answer user specific questions are:

a. Log Likelihood – to measure the importance of query terms and assign term and document weights.
b. Summarisation techniques - to extract sentences relevant to the user query.

H4. Topic signatures can be acquired and used for computational tasks, using local analysis techniques and statistical weights without the intervention of an expert user.

H5. Using the Question Answering and Automatic summarisation techniques, students will be able to get the correct answer quicker and looking in fewer places than using standard search engines.

1.5 Originality of work

Develop a Question Answering system using statistical methods

A set of algorithms is proposed in this work that can provide better or similar quality retrieval results as a baseline search engine but reduce the amount of irrelevant information the users receives from the system. The metric mainly used to measure surprise usage of terms and their weight is Log Likelihood. In cases where lower accuracy is required of the statistical measure, TF.IDF is used. To understand the query, our system uses a stop word list for quickly filtering the question terms and then normalises the weights of each term in the query using what is described as local analysis techniques so that each term has a unique weight depending on how "important" is identified by the frequency in the domain. To pick the correct document, there are a few different approaches available, like the sum and average of the statistical weights associated to important. Finally an algorithm that depends on the frequency of the terms in sentences that contain the keywords is used to extract the answer from the selected document.

Although the metrics used are widely used in Information Retrieval / Information Extraction applications, the method described above has not been used in order to answer questions within a Virtual Learning Environment. Also, there are enhancements in the techniques used which are described below.

Acquire Topic signatures automatically

Previous work in topic signatures, demonstrate extraction techniques using human intervention. While the actual weights of the signature terms were calculated automatically, the signature terms were picked by a domain expert. In this work, we use global techniques in order to identify topic terms for the domain. Picking up signature terms is also a semi manual process in the work we have seen so far. In our work, we identify a threshold for each potential signature term using statistical metrics and develop the topic signature in a fully automated way.

Use summarisation techniques to retrieve answers

Document Summarisation and Question Answering are generally two areas with many similarities. A widely used approach to document summarisation is to identify the main sentences of the document and return them as a summary. We took this approach a bit further by weighting sentences depending on the query.

A simple algorithm was also developed in order to include sentences that are useful in the answer but do not carry any special weight and also if more clusters were identified as potential answer, to be able to pick only one as the final answer.

The challenges in a Question Answering system come from multiple factors and identifying the one correct set of sentences that will answer the right question is a difficult task. To add to this the Virtual Learning Environment introduces more challenges.

One of the main challenges is the dynamic nature of the content available in the Virtual Learning Environments. The content of a Learning Environment will vary from time to time, so identifying any answers should work the same in smaller corpuses and larger ones. Also the preference we have for statistical methods will lead to some challenges. Using statistical approaches removes the semantic information, information that can be extracted from existing knowledge bases and also information from the syntax and morphology of the text. This is a major challenge, because usually the above knowledge contains important information that can be used by a Question Answering system.

Finally the lack of baseline Question Answering system makes the task of evaluation harder. We have set up our evaluation in a way that we can test components individually and compare them with baseline systems, using domain knowledge and finally run a user test.

1.6 Data

There are a few sources of data that have been used during our development and evaluation. First of all, we have the main corpus that the answers are retrieved from. For this we used the CCNA online notes. The content files were also made accessible to the users, via a baseline search

engine interface in order to conduct the user experiment described in section 4.3.

At the end of each chapter, the CCNA notes contain online self-assessment tests where the user can check their knowledge. Cisco provides the expected answer for each question. From these self-assessment questions, 10 questions were used in order to test the efficiency of the algorithms in the different runs as described in section 3.The same questions are used as part of the user experiment, described in section 4.3.

A reference corpus was also used, which is based on the British Academic Written English Corpus from Oxford University (Nesi, et al.2007) in order to derive statistical metrics for the statistical calculations. This corpus was used as a reference for comparing the frequencies of terms that are found in the CCNA notes in order to identify any overuse or underuse in the frequencies of the terms that appear in both corpuses. The underlying assumption is the fact that terms relative to a domain will appear more often in documents describing the domain.

The final data resource we used was a stop word list. The stop word list we used was the one distributed by Princeton University (Sedgewick, Kevin, 2008).

1.7 Data preparation

The CCNA offline package needed to be pre-processed. The first cleanout exercise removed all the HTML code from the files. The second phase of pre-processing has split the files into documents, sentences and words and

also preserves the links between these lexical entities. The total number of files processed by the system was 246.

The questions were also slightly modified since the CCNA self-assessment questions are multiple choice type. So for example if the question was “which of the following are functions of a router in a network?” we have transformed it to “Which two functions does a router perform in a network?” The answers from the CCNA questions were also as comparison point with the answer provided by the system.

Finally the reference corpus is provided in clean text format, which did not need any pre-processing.

1.8 Thesis outline

The remainder of the thesis looks into the development of a Question Answering system. Chapter 2 contains a literature review of the state of the art technologies used in Question Answering Systems.

Chapter 3 contains implementation details of the system. The different modules are described in detail and developed to support the hypothesis testing for this project. This chapter also describes the algorithms used for each stage of the Question Answering system. This description is followed by an explanation of the pilot runs (section 3.9) detailing how each pilot run was run explains how we run each pilot and if any enhancements were required to improve the algorithms and any subsequent runs. The rationale supporting the choice of each algorithm is also explained in the following section. Finally a separate section captures the limitations of our system

and the chapter ends with a section on hardware and software requirements for our experiments.

Chapter 4, describes the evaluation process and is split into two main section. The first section describes the evaluations that were performed during the development of the Question Answering system. The second part of the chapter presents the user evaluation and more specifically the experiments conducted with a group of MSc students at the University of Sunderland. In this chapter, the results of the questions the students were asked are presented alongside with any metrics captured like the time spent on answering a question.

Chapter 2

2 Relevant Literature

2.1 Introduction

In this chapter we will report the techniques that underpin the research. The first part of this chapter contains a general overview of different question answering systems and the technologies they use at each processing stage. We group the functionality provided by a Question Answering system in three main areas – Question Processing (QP), Document Retrieval (DR) and Answer Pinpointing.

In the next section of this chapter three main areas of modern Question Answering system are considered with statistical Natural Language Processing (NLP) methods that have not been used in QA systems but perform well in other areas of NLP.

2.2 Question answering systems

When we ask a question to another person, the person initially listens to the question and tries to understand what we are asking them for. Then the person tries to recall previous or combined knowledge in order to form an answer. Once all the background information is collected, then the person replies to the question.

Similarly in Question Answering Systems, there are three phases. The first step, which we will refer to as Question Parsing, is when the user enters a query into the system. The system in this phase tries to extract as much information as possible for the query that would help identify relevant documents on the next phase. Once the information is extracted, documents that score highly on a relevance metric are picked. This second step will be referred to as Document Retrieval in this report. Finally, from

the selection of relevant documents, an answer is formulated and returned to the user. This final step is referred to as Answer Pinpointing in this report.

In late 70's the first QA systems such as *STUDENT (*Bobrow, D.G 1964) and *Lunar* (Woods, W.,1973) were developed as interfaces to problem-solving systems. Still nowadays, QA systems are being employed as interfaces to expert systems using large databases and reasoning mechanisms. Also there is an active need of a move from the traditional search engine to Question Answering systems in the cases where the user would want specific information. Google dominates the market of Information Retrieval but a big proportion of the modern web user that spends half an hour a day searching is moving to systems like Ask Jeeves (Roussinov , Fan, Robles-Flores, 2008).

Question Answering systems can be categorised into two main types depending on the technologies used. These two categories correspond to the linguistic approach and the statistical approach. There are some systems that use a mixture of the technologies, but they do fall into one of the main categories depending on which approach is used predominantly.

In the linguistic based approach we have systems that use external knowledge and various tools such as named entity taggers, WordNet parsers (Prager, Chu-Carroll, Czuba, 2001), some manually annotated corpora and ontology lists (Xu, Licuanan, Weischedel, 2004) for answer pinpointing. TREC evaluations have scored highly systems that used shallow NLP techniques for the process of identifying the correct answer and systems based on Text Patterns (Ravichandran, Hovy, 2002)

(Soubbotin, Soubbotin, 2001), but also systems that are built around the data that the corpus contains using web queries.

In the statistical approach the main advantage is that there is minimum or no pre-processing required, so a large amount of data can be used as the corpus and also the data can be in any morphological form. Statistical approaches do perform well when an appropriate size of data is available to perform the calculations. The corpus can be updated when needed and the statistical algorithm can be run over the updated corpus to update the measurements for each of the terms in the corpus. The main drawback of statistical question answering systems is that by not using the linguistic features of the words, terms are treated independently or as a part of an n-gram.

Methods that are used in statistical analysis include SNV classifiers, Bayes law, TF.IDF and other statistical measures and techniques. What is researched with statistical methods is how to overcome the limitations introduced by not using any syntactic or linguistic information. Also statistical techniques try to make the system more responsive to updates in the corpus where an NLP system would require updates to their knowledge base.

Regardless of the type of technologies used, statistical or linguistic, the Question Answering problem can be treated as a multi-step task. Some systems may split the major steps into sub steps but typically a generic structure of Question Answering systems is adopted. We will provide more details in the next section.

2.2.1 Structure of QA Systems

A general framework for Question Answering systems consists of multiple modules that work in serial order to return the answer to the user. The input from the user is usually a question in natural language format. This input needs to be initially “understood” by the system. The first module of many statistical QA systems creates a query from the user’s question. We will be referring to this part of the system as Question Parsing (QP) module. The main responsibilities of this module is to process the input text to a format that would be appropriate for the rest of the system.

The next module would rely on the output of the QP module and will use it in order to retrieve potential documents that would answer the question and we will refer to it as Document Retrieval (DR). The Document Retrieval techniques we will investigate in this thesis are strictly statistical. The output of this module usually consists of a list of documents, passages or sentences that can be used to answer the question together with an associated set of values that can be used to rank the document in the list.

The final state that QA systems go to is providing the answer back to the user. We will refer to this module as Answer Pinpointing (AP). This module will receive as input the list of documents and then try to identify the answer that is returned to the user. Different techniques have been used in various systems which we will describe in the Answer Pinpointing section. The different techniques that are being used I by different systems are going to be described in the following sections.

The systems are used to describe the different stages of a question answering system are AquaLog (Lopez, Motta, 2004), AskMSR (Brill et al. 2002) and various other approaches.

There is the possibility that one part of the system consists of multiple sub-parts, and these are described in the corresponding section with further details on the implementation.

2.2.1.1 Question Parsing

The basic idea behind question parsing is the same across all QA systems and this is to transform a natural language query submitted by a user into a representation that the system can understand and process. The differences in the approaches derive from the need to create the query an as-general-as needed manner in order to retrieve all the possible information from the knowledge base or the corpus. At the same time one should avoid over-generalisation so the terms used can retrieve information to satisfy the user query.

At this stage the QA system will have to determine the question type and also the type of answer that is expected to answer the specific question. For example, if the question is “Where is the river Wear?” the user will expect a location to be returned from the system. Knowing that, we can filter out non location expressions from the corpus or the knowledge base that refer to the river Wear (e.g. river Wear is polluted). Another issue that arises is when the corpus contains passages such as “St. Peter’s campus on the bank of river Wear” which shows that Wear is near the St. Peters campus. Having only this information available on the corpus, we need to

identify where St. Peter's campus is and then return the more general location if the user requires so.

In AquaLog (Lopez, Motta, 2004) some of the functionality required above is being done using an ontology. This QA system, receives the ontology and the user query as the input and then processes the rest of the information. The data model that AquaLog uses is a triplet one. So the initial process module transforms the user query into a triplet of the form <subject, predicate, object>. The choice for that is because in practice most queries can be represented in binary relational model. Also most semantic web schemas support this triplet data model.

When it comes to applying this in an e-Support system for learning environments, the backend ontology should be able to handle all possible user requests, so we need to have a dynamically maintained/acquired ontology to meet the requirements of the institution's Virtual Learning Environment. A potential problem may arise when the triplet based data model is not enough to cover the user's needs which will be when the input query is not a factoid one, and requires more complex processing. In the AquaLog architecture, if the user's query cannot be transformed into the triplet binary format, the system requires the user's input in order to clarify and reconstruct the question. In a learning environment such approaches should ideally be minimal or even better avoided. Since the system would aim to support students through their learning activities, query reconstruction may be time consuming for the learner and also take the concentration of the student away from the initial task. Also in cases where the query is too complex to be represented by a data triplet then the system

will fail to respond to the learner. In research on discussions forums Donghui, F, Shaw E., Jihie, K., Hovy E. 2006), it has been found that complex answers cover a big proportion of discussions students ask academic staff.

Another example of question recognition can be taken from Cao et al. (2005) where a question template is used in order to return the correct answer to the user. The data structure of the question template contains information such as the type of the expected answer, the focus of the question which is the core noun, any persons or named organisations in the question string, the key verbs, and also any instances of location, time or numeric values that can be identified within the question. Patterns are being used in (Brill et al. 2001) as well to parse the questions although in this case patterns are manually created for the specific project. The output of the parse contains a 3 tuple data structure in the form of (string, L/R/-, weight) where the string is the reformulated query, the next tuple represents the location where the answer could be found (Left (L), Right (R), Anywhere (-)) and the weight tuple represents the preference that the system has in finding answers using the reformulated query. A higher precision query string can be “Abraham Lincoln was born on” where a lower precision one can be “Abraham” “Lincoln” “born”.

Cao,J., Roussinov, D., Robles-Flores, J.A., Nunamaker J. (2005) gives another example of parsing the input question and that is with the use of patterns. The use of patterns is also being implemented in Answering Definition Questions Using multiple Knowledge Sources (Hildebrandt, W., Katz, B., Lin, J. 2004) where the target term of the question is extracted

using regular expressions. A list of patterns is stored within the system and the question is parsed. If the question does not fit with any of the stored patterns, simple heuristics are applied to the question in order to extract the target part of the question. Similar approaches (Cao et al. 2005) have been using semantic similarity algorithms such as Latent Semantic Analysis to measure the similarity of previously asked questions in order to check if an answer already given to a question will fit to the user's query.

Soricut and Brill (Soricut R., Brill E., 2006) in their work used a statistical chunker in order to transform the question into a query. The chunker uses Dunning's (Dunning, 1993) log likelihood in order to identify any 2 or 3 word co-locations. The log likelihood measures the probability of a co-location to occur in the answer or the query compared to normal usage of the collocation. This builds a bridge between the query and the answer that is not there in statistical approaches due to the luck of structure in the answer. The log likelihood is calculated, for each term (it can be multi word terms) of the query. If the term is a unigram, the chucker assigns the score of 1 and if it's a bigram or trigram the chunker assigns the log likelihood value to the term. So the end outcome is a query with weighted terms, where more important terms would have a greater weight.

Another QA system that was built with a VLE usage in mind was presented in 2005 in the Journal of E-Learning (Kumar et al, 2005). In the Question Parsing stage, the system is using a Link Grammar Parser to identify the syntactic structure and retrieve the verb and the noun phrases (Temperley, Sleator, Lafferty, 1993). There is a sub-module in the Question Parsing phase which is using an Entity recognition considering the output of the Link

Parser. The table of contents of each document or the headings and sub headings are used in order to pick up named entities which should be either noun, verbs or adjectives. For the final step of Query Parsing the system does a query formulation by adding weight of 2 to the object and verb identified by the first sub-process and then assigns weight of 0 to the stopwords. Finally the rest of the words get assigned the weight of 1 and there is also some query expansion (Gonzalo, Verdejo, Chugur, Cigarran, 1998).

Support vector machine (SVM) classifiers have been used in a few statistical systems like Moschitti's (2003) to classify question categorisation. Zhang and Zhao (2010) has also used SVMs in order to classify questions. Zhang's version had to overcome some limitations of SVMs which are basically binary classifiers. The limitations are that the number of samples has to be fixed and also the model needs to be retrained each time a new sample is added. They used a similar approach as we did with a large corpus to use as reference so the only part of data that can change is the actual system data which is fairly smaller or can be smaller in a VLE environment that can use teaching modules/courses as corpus boundaries. Zhang and Zhao mentioned that SVM can also be used for question extraction, but because of the imbalanced sample numbers of answers and non-answers an improved K-mean algorithm combining voting for answer extraction was used. (Zhang, Zhao, 2010). In Moschitti's implementation, the document weight is calculated according to the following formula:

$$w_f^d = \frac{l_f^d \times IDF(f)}{\sqrt{\sum_{r \in F} \left(l_r^d \times IDF(r)\right)^2}}$$

Where

$$l_f^d = \begin{cases} 0 & if\ o_f^d = 0 \\ \log\left(o_f^d\right) + 1 & otherwise \end{cases}$$

and

$$IDF(f) = \log\left(\frac{M}{M_f}\right)$$

The weights for each document will place the document in a specific place in space. When the algorithm runs in training mode clear boundaries should be configured between documents of different categories. When the application is running in Question Answering mode, an SVM works like a binary classifier, taking into consideration the categories identified in training mode.

More recently in 2010, a Chinese question answering system (Zhang, Zhao, 2010) uses classification in order to process a user's question. Some key categories like Time, Location etc. have been identified and features have been added to each category for example the Time category would have as features the Year and Month. For the classification this system uses POS, Named Entities, semantics and the words of the sentence, which can make it less adaptable since POS taggers are language specific and Named Entities need to be compiled individually and updated regularly to reflect the corpus. Purely statistical implementations for categorisation have also been implemented in (Soricut R., Brill E., 2006) but we will look

more into them later in section 2.3.2 since they are part of a different module present in the generic structure of Question Answering systems we defined in 2.1.1. This work uses four different methods for categorising a question into pre-defined categories. These methods are Boolean, TF.IDF, Entropy based weighting and semantics for questions. The Boolean weighting assigns 1 if a feature is present and 0 if the feature is not present in the data. In the TF.IDF approach, the feature value is given as

$$w_i = \log \frac{N}{TF * IDF}$$

Where N is the number of training samples, IDF is the inverse document frequency measuring the importance of a term in a document set and TF is the term frequency in the document set.

The entropy based approach gives a value to a feature by using the formula:

$$H_{(i)} = \frac{1}{\log N} \sum_{k=1}^{N} \left[\frac{f_{ik}}{n_i} \log \left(\frac{n_i}{f_{ik}} \right) \right]$$

(Zhang, Zhao 2010)

where f_{ik} is the frequency of a feature i in category k, n_i is the frequency of a feature i in the collection of samples and N is the total number of samples. The approach assumes that if a feature distributes evenly through the samples then the entropy reaches the minimum.

With such conditions, the value of the feature would be

$$a_i = \left\{1 + \frac{1}{\log N}\sum_{k=1}^{N}\left[\frac{f_{ik}}{n_i}\log\left(\frac{f_{ik}}{n_i}\right)\right]\right\}$$

The value of the feature would be less the greater the entropy is which indicates a feature would be less important.

The semantics approach generally uses two methods. One is to manually increase the semantic annotation of the nouns, whereas the other relies on a semantic similarity matrix of words and a feature vector. The new feature vector would be A' = A *B where A is a feature vector and B is a semantic similarity matrix.

In Jun Suzuki's (Suzuki J., Sasaki Y., Maeda, E. 2002), a collection of features is selected in the Question Parsing (in the specific project it is called Question Analysis Module). This collection of features includes the keywords of each question, the type of question, any numerical units and auxiliary terms.

2.2.1.2 Document Retrieval

Once the question has been parsed and understood by the QA system, the next step is to retrieve passages of text or full documents that will be used to return the answer to the user.

Revisiting AquaLog (Lopez, Motta, 2004), its backbone has a relation similarity service (RSS) which tries to match the term relation output which has already been classified to a question type. Initially an attempt is made to match the parsed question string with the ontology as well as the information stored in the Knowledge Base (KB). Further examination of the

question query is carried out using techniques such as string matching, and making use of lexical resources. For example when the user asks the system “who works in the semantic web”, the RSS will identify that the semantic web is a research area in the KB. Also from the question type, RSS will need to find a link of a user or organisation to return since the input query is a WHO type question. The next step is to try to identify the relationship of the ontology where in this case the only relationship in the ontology is the has-research-interest. AquaLog will then return this relationship to the user and wait for the user’s input to verify that this relationship is the desired one. This approach does not comply with the TREC (Is question answering a rational task?) question answering track specifications which allows QA systems to accept as input only the user query. Since our approach would be for learning environment systems, neighbour concepts retrieved from the query can be displayed to the user in case there is any area that needs to be investigated by the user. Similarly, in Pasca et al 2001 a taxonomy is used at the core of the Question Answering system retrieval. The main hypothesis behind this system is that the passage that will contain candidate answers will not only contain some of the question keywords, but also a concept of the same semantic category as the concept inquired by the query string.

In pattern based systems such as Xu, Licuanan, Weischedel (2004) and Cao et al. (2005) the answer extraction is generally based on pattern matching. The storage of the knowledge base varies from system to system and can be an indexed database, a link to a dictionary or other lexical resource, any corpus built to support the system, and of course the web. In

most cases in the retrieval process answer type classes are used, that indicate the desired answer. When retrieving passages, a slot filling algorithm is being executed to transform free text into answer types.

An interesting and more sophisticated approach comes from Hermjakob, Hovy, Lin (2003) which although rely on IR techniques which have been successful in QA systems, provides an additional interesting feature. Their system contains the CONTEX parser that can add some external knowledge to the system and the query string can contain information such as Logical-Subject and Logical-Object. Using such information in the query string and to constrain the potential answering strings, the matching of the right answer becomes easier for the system. When a user is looking for types of variables in Java, having external knowledge of the real world incorporated will make the retrieval of relevant passages easier since the QA system will be able to determine that Java is a programming language. On the other hand if several keywords are retrieved in a document or passage, the chances of that document to be irrelevant to the query are low (Sparck J., 1998).

Statistically based approaches have also been used to retrieve information from the corpus. The discussion board bot (Donghui, F., Shaw, E., Jihie, K., Hovy, E., 2006) that replies to student queries uses term frequency and inverse document frequency to retrieve relevant paths. The hypothesis behind this approach is that any passage found with similar words to the query will have some semantic relation with the input query which means that it will possibly be of the user's interest. Cosine similarity is also used in this system to retrieve similarities between the query posted and passages

in the text. At this point the passages are pre-processed into semantically related tiles so that each document used in this approach would contain an average of 10 semantically different tiles. In AskMSR-A and AskMSR-B (Brill et al. 2002) n-gram harvesting is deployed in order to extract passages that are relevant to the user query.

N-grams have also been used to score sentences in the training set of the system developed by You Ouyang, Sujian Li, and Wenjie Li (2007). The hypothesis behind the usage of human summaries to compare sentences from potential answers. The closer the sentence was to a human summary, the higher it scored. The comparison was done by calculating the frequency of a single n-gram in one summary and also the maximum frequency of the n-gram in all summaries and also the average frequency in all the summaries.

In Zhang, Zhao (2010), document retrieval occurs with a previous step of processing the answer sentence. Initially sentences are picked from the corpus using an open source search engine (Lucene) and the sentences of each document are pre-processed to identify if there are any keywords or named entities in the sentence.

Arvind Agarwal et al (Agarwal A., Raghavan H., Subbian K., Melville P., Lawrence R., Gondek D., Fan J., 2012) described some fundamental differences between the retrieval techniques required from a Question Answering system in comparison to search engines. The differences they used in their system, was that instead of relevance, they used binary relevance judgements to state if the answer is correct or not. Another

difference is that in search engines, documents have different degrees of relevance to a query, where in QA systems, the answer will be included in one or a small proportion of the document.

In Proceedings of the 21st international conference on World Wide Web (Unger C., Bühmann L., Lehmann, J., Ngonga Ngomo A., Gerber D., Cimiano P., 2012), in order to rank the documents retrieved, two different scores are used. A similarity score based on string similarity and the prominence score. The two scores are combined using a learning function with the impact of similarity and prominence being controlled by a function variable. Although the system is using different techniques than the ones we investigate in this thesis, the idea of combining different metrics depending on the impact they have on the Question Answering task is something widely used within our system.

One of the most impressive Question Answering system of our time is IBM's Watson. Ferruci (2011) explains the approach they took in building the system. The document retrieval part of Watson consist of the "primary search" stage which uses different retrieval techniques (search engines, SPARQL etc.)to collect as much data as possible for the question. The data collected is then passed through the "Candidate Answer Generation" module, which is mainly using morphological approaches in order to create potential answers to the question. We need to stress that at this stage the correct answer must be included in the list of candidate answers. The following processing stage, filters the list of candidate answers and passes the filter list to what is called "Hypothesis and Evidence scoring". The evidence scoring module includes metrics similar to local analysis

techniques and also data from triple stores. Our system uses a similar technique in the form of topic signatures.

Another multi feature approach comes from (Surdeanu M., Ciaramita M., Zaragoza H., 2011). The features used in this approach are:

- BM25 similarity feature that uses the term frequencies of a question term i in the question and potential answer, uses a length normalisation variable and the inverse document frequency
- Translation feature, to enhance the bridge between the lexical chasm between questions and answers. Similarity only based model may suffer in identifying an answer that is represented using different words from the question.
- Density and Frequency Features, where the order of keywords identified in the question and answer, the answer span, number of non-stop words are also used as a potential feature to rank the answer.

So far we reviewed techniques used for the second main part of the Question Answering system, in both statistical and linguistic approaches. Both techniques have advantages and disadvantages when used to perform the task. The next section moves to the final step of the Question Answering task, where the answer is being picked from a collection of documents and presented to the user.

2.2.1.3 Answer pinpointing

The third stage of a QA system is the one that will determine the precision of the system. Although initially document retrieval will return a collection of documents that contain the right answer, the unique answer that will be

returned to the user will depend on the initial natural language query and the weighting algorithm that the system uses to pinpoint the answer. One of the main problems, in this state, are ambiguous potential answers. Different approaches to overcome this have been applied by different systems. AquaLog uses the user's input in order to override this. For example, if there are more than two ontological categories as potential solutions to the user's query, areas of disambiguation can occur at different stages of the question answering lifecycle. Initially, it can occur when the user's query is too general for the system to determine one solution with higher ranking. For example if we ask a QA system "Why John is famous", unless the corpus limits to 1 instance of "John", multiple answers will occur.

Another issue with answer pinpointing is corpus limitations, which can occur when the data we provided the system does not contain the relevant information in a readable format for the information retrieval (IR) engine. For example, if the corpus contains the following passage is available "Belli's clients have included Jack Ruby, who killed John F. Kennedy assassin Lee Harvey Oswald, and Jim and Tammy Bakker.", it will be difficult for the IE/IR engine to identify that "Jack Ruby" did not kill Kennedy but Oswald did (Hermjakob, Hovy, Lin, 2003). Hovy, Kim, Shaw and Feng (2006) return the answer that is the closest to the user's query since the corpus is limited to one specific course material category. This may be the most appropriate technique for most academic based QA systems, since we don't want to have any dependencies with manually built knowledge derived from the nature of the corpus. However there will be some restrictions on what kind of answers the system will be able to give and this

will be bound to the learning objects that each institution will have at each time.

In Hermjakob, Hovy, Lin (2003) the pinpointing and weighting of the answers is being done in a way that different heuristics are applied to give the answer the appropriate weighting. For example greater weighting is given to proper names returned from the IR engine, if an upper case matching with length more than 1 is being done thus is assigned an extra weight. Also there are discounts applied to the document if an external source has been used to justify its inclusion in the result list, such as WordNet synonyms, stemming matching, etc.

A different approach is adopted by (Brill et al., 2003) and (Ravichandran, and Hovy, 2002) who pass the candidate answer to an IR engine accompanied by the keywords extracted from the user's query. A best match algorithm is applied to the documents and the first document returned will be the answer returned to the user. There is an extra feature in Brill et al. (2001) that if a document containing a candidate answer is returned to a different query, this answer will be preferred by the QA system since candidate answers tend to be related to the correct answer, and multiple occurrences of a document suggest that the document contains either the answer or terms related to the answer.

The QA system presented by Soricut and Brill (2006) is uses two different techniques in order to extract answers from documents. One technique is based on n-gram co-occurrence, and the other on automatic translation techniques. The n-gram technique assigns a weight of 1 for each unigram,

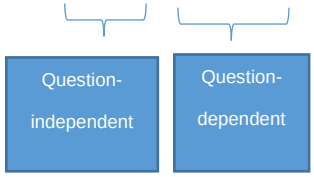
and a weight equal to the likelihood ratio for each bigram and trigram found in the input question that has a likelihood ratio greater than 1 as computed from the corpus used to train the algorithm.

The translation inspired techniques uses a variation of Bayes law as shown in the formula below.

$$p(a|q,T) = \frac{p(q|a,T) \cdot p(a|T)}{p(q|T)}$$

The denominator of the formula above can be ignored since it will be a static

$$a = \arg max \; p(a|T) \cdot p(q|a,T)$$



To weigh the best answer in the system by Zhang and Zhao (2010), a collection of metrics was used which then were passed through a k-means algorithm. They are:

- Quantitative features: which reflect the ratio of query words matched in the sentence
- Density: that is calculated from the formula :

$$Density = \frac{Minumum\ windows\ contain\ all\ of\ mached\ words}{Length\ of\ window}$$

  - Sequence features: Which are the measures of similarity of word sequence between words that are matched both in answer sentence and the query. This feature receives a weight of 1 each time a term is matched.

For example, when applying Zhang and Zhao (2010) features for the query words **network interface card** (**NIC)**, the weights in the table below are assigned to the sentences.

Candidate sentence	**Feature weight**
This page will explain how an adapter card, which can be a modem or a **NIC**, provides Internet connectivity	1/4
A NIC provides a **network interface** for each host	2/4

  - Another sequence feature is the similarity of the word sequence between matched words in answer sentence and the question
  - The final sequence metric uses the ratio of the total content terms in the question (without the stopwords),in our example 4, over the content words in the potential answer – 9 for the first candidate answer (*page, explain, adapter, card, modem, NIC, provides ,Internet, connectivity*) and 5 for the second one (*NIC, provides, network, interface, host*)
- The selection of the final answer from candidate answers is done by using all the features above into a vector and applying a k-means algorithm

In MULDER (Kwok et al., 2001), each potential answer is tagged as summary. For each of the summaries, the distance between each summary and the keywords of the query is calculated. Two values are compose the final weight, K_L and K_R.

K_L, is the sum of all the keyword weights on the left, thus the L, of an answer word over the number (m) of the unrelated word on the left side of the answer word, as show in the formula below:

$$K_L = \frac{w_1 + ... + w_n}{m}$$

K_R, is the sum of all the keyword weights on the right of an answer word over the number (m) of the unrelated word on the right side of the answer word. The final score for a candidate answer is *max(K_L,K_R)* (Kwok et al., 2001), but specific type of questions can have K_L and K_R modified with multipliers if there is likelihood of the answer to be on one side over the other.

In this section we identified techniques that are used by Question Answering systems to solve the task of Answer Pinpointing. These techniques include statistical and linguistic approaches. In the next section, we will concentrate on specific techniques that we used during the research in order to develop a statistical algorithm used within a VLE to answer user queries.

2.3 Statistical Natural Language Processing (NLP) techniques relevant to Question Answering (QA) tasks

In this section, we include relevant literature, describing statistical technologies used to perform the three main tasks of the Question Answering system. The section is broken into sub sections that map to the three processing stages of a question answering system. Section 2.3.1 contains statistical techniques that were used in the Question Parsing tasks in other systems. Section 2.3.2 contains information about statistical weights that can be used for Document Retrieval. Section 2.3.3 contains information on how topic signatures can been exported from a corpus. Finally section 2.3.4 contains techniques used for Sentence extraction.

2.3.1 Query Expansion using Local and Global Analysis for the Question Parsing Task

Query Expansion is a technique that can be used in the QP task of a question answering system. The idea behind Query Expansion (QE) is that before submitting a query to an information retrieval engine, we augment the query with terms that are not part of the original query but are contained in the original corpus. The need for query expansion arises from the nature of keyword based searching. The user will enter a short query into a system in order to retrieve some documents. Unless the query contains topic specific keywords (Carpineto C., Romano G., 2012), there is a big chance that the query is ambiguous. Local and global analysis are two techniques used to expand queries for Question Answering or Document Retrieval tasks. To define the two methods, local analysis uses only the top ranked documents to expand the query, whereas global analysis uses the full available corpus in order to pick the expansion terms (Carpineto C., Romano G., 2012).

2.3.1.1 Local Analysis

Local analysis is based on using data from the top *n* documents returned by a query in order to identify potential terms to expand the original query (Xu, Croft, 1996). Local analysis has two slightly different methods, local feedback and local context analysis. In local feedback, the top documents returned by a query are used to build a thesaurus of the query terms. The probabilities of term occurrence are then used to give different weights to query terms. Unlike global analysis, this technique does not add any more terms to the query. Local context analysis on the other hand despite the name, combines techniques from local feedback and global analysis. In

local context analysis, a term is passed to an IR system and the top n ranked documents are being retrieved. The documents in this case are like the pseudo-document in global analysis, a window that surrounds the concept. The concepts within the top ranked documents are weighted using a variant of TF.IDF and the top m concepts from the pseudo-document are then added to the query.

The main disadvantage of the local techniques is the overhead created by using two queries on the corpus. One is used to set up the weights (Xu, Croft, 1996) or to add terms in the query (Local Context Analysis) and the second query to the IR system aims to retrieve the document related to the enhanced query.

On the other hand, the work done by Lam-Adesina and Jones (2001) used summaries for query expansion. This was motivated by the hypothesis that expansion terms should be picked from the most relevant parts of the document (Lam-Adesina and Jones 2001). The query based summarisation task is very similar to the QA task and expanding the query with relevant terms will only increase the weight of documents that contain the relevant information. The sentence selection task for the summary is a variation of local analysis, where instead of using the top n-documents to make the expansion the system is using a summary of the document. The summary does not need to be created at the same time as the user query and it can be cached on the persistence layer of the application in order not to add to the overhead. The features used to weight the sentence significance to select sentences that should be added to the summary in Lam-Adesina and Jones (2010) work are:

(1) sentence position within the document;
(2) word frequency within the full-text;
(3) the presence or absence of certain words or phrases in the sentence;
(4) a sentence's relation to other sentences, words or paragraphs within the source document;

The system is evaluated by passing the same queries to a baseline system- without query expansion, to a system with global techniques and also systems using multiple versions of the summary based local context. Local techniques provided the best results, with an average 10% improvement on standard selection.

2.1.1.1 Global Analysis

Global analysis is a technique used initially for query expansion (Xu J., W. Croft W.B., 1996) and later a variation of it Local Context Analysis for information retrieval (Xu, Croft, 2000). Global analysis uses the full corpus in order to expand the query. Word co-occurrences and relationships of terms are used to perform IR related tasks. On a task of identifying n-grams in a query using a technique like that would produce results similar to the ones for query expansion. One of the disadvantages of this technique is the processing overhead for identifying the *n-grams*. In the literature Qui and Frei (1993) used a global analysis technique. The authors excluded the stop words from the corpus and used every other word as a concept. The words that co-occur with a word interpreted as a concept in the same document are the context of the word. Crouch and Yang (1992) used global techniques with clustering to determine the context for document analysis. As mentioned above in global analysis every non-stop word or sometimes only nouns are considered as concepts. A pseudo-document is associated

with every concept in the corpus. This pseudo-document contains all the words that co- occur in a static window around this concept in the corpus. When used in query expansion, the top ranked concepts from the pseudo document are added in to the query in order to expand the query terms.

Global or corpus specific techniques are much faster than local ones since there is no overhead of running a query to retrieve the top documents, but they do not perform as well as local analysis (Carpineto, Romano, 2012).

In this section we described some techniques used in the query expansion task of a Question Answering system. These techniques have inspired the research described in this thesis and are used and expanded in the development of the algorithm we describe in section 3. The next section describes statistical weights used in the algorithms developed in the system. These algorithms are used throughout the development of the system.

2.3.2 Statistical Weights used for Document Retrieval

There are two main statistical measures used to assign weights to terms and bigrams within the Question Answering System developed and described in this thesis. These measures are also widely used in the Information Retrieval literature and are known as TF.IDF and Log Likelihood.

The TF*IDF measure is calculated multiplying the term frequency (TF) with the Inverse Document Frequency (IDF). TF can be calculated by dividing the number of times a term appears in the document by the amount of terms that are available in the document. The IDF is calculated by getting the

logarithm of the total number of documents divided by the number of documents that contain the term *w*. For example for term w in document d TF*IDF is calculated as

Equation 2.3-1

$$\textbf{TD*IDF}_{\textbf{w,d}} = tf_{w,d} * \log\left(\frac{N}{df_w}\right)$$

Where $tf_{w,d}$ is the number of occurrences of term *w* in the document df_d is the number of documents containing the term *w* in the collection *N* of documents.

For the log likelihood feature, Dunning's definition was used (Dunning, T. 1993). The main idea behind the log likelihood metric is to calculate the "surprise" of an event occurring more than usual.

The log likelihood feature can be calculated with the formula below:

Equation 2.3-2 –Log likelihood formula

$$G = 2 \times \left(\left(freq_{domain} * \log\left(\frac{freq_{domain}}{freq\_Expected_{domain}}\right)\right) + \left(freq_{general} * \log\left(\frac{freq_{general}}{freq\_Expected_{general}}\right)\right)\right)$$

where the expected frequencies can be calculated from the formulas below:

Equation 2.3-3 – Expected frequency (Domain)

$$freq\_Expected_{domain} = size_{domain} \times \frac{freq_{domain} + freq\_general}{size_{domain} + size_{genereneral}}$$

Equation 2.3-4– Expected frequency (General)

$$freq\_Expected_{general} = size_{general} \times \frac{freq_{domain} + freq\_general}{size_{domain} + size_{genereneral}}$$

Finally the log odds feature is calculated using the following formula.

$$logodds(wc) = \log \frac{f(wc)f(\bar{w}\bar{c})}{f(\bar{w}c)f(w\bar{c})}$$

Where:

$f(wc)$:is the frequency of word w in the collection c

$f(\bar{w}\bar{c})$:is the frequency of other words in the collections except c

$f(\bar{w}c)$:is the frequency of words other than the one the feature is calculated for in the collection

$f(w\bar{c})$: is the frequency of the word the feature is calculated for on the collections other than c

In more recent Question Answering system like Freebase (Yao X., Van Durme B., 2014), use statistical approaches were proved essential in order to enhance the performance of the Question Answering system, which is based in linguistic approaches. Freebase, uses relationships in a knowledge base in order to answer questions. One of the main problems faced by the researchers, was that the relationships formally defined in the knowledge base may not be natural language friendly. So for example, the relationship *brother/sister* would be defined as *sibling* in the knowledge base. For the Question Answering system to be able to map the knowledge base relationship *sibling* with the query term *brother* (or *sister*), the system needs to have a list of sub relations based on natural language. At this point we should mention, that each formal relationship has some arguments that define it, e.g. *Siblings (Person – Person)*. To accomplish that, an external

corpora is used where initially the relationships based on the knowledge base are extracted. The next step of the process is to extract sub relationships and then map them to relationships from the knowledge base. The extraction step is performed using statistical approaches, in order to identify potential alternative sub relations that use the same arguments as the knowledge base relationship.

Once a list of sub- relationships is extracted, in the specific paper 1.2 billion relationships, these sub-relationships need to be aligned with knowledge base relationships. For the mapping process, IBM alignment Model 1 (Brown et al., 1993) was used. The introduction of statistical processing in this mainly linguistic approach, increased the F_1 score from 39.5 to 44.3

To support the hypothesis that log-likelihood improves the results of a Question Answering system, Heie, Whittaker and Furui (Heie M., Whittaker E., Furui S., 2010) developed a system based on the model that the probability of an answer A for a question Q depends on the A depends on two sets of features: W = W(Q) and X = X(Q) where W represents a set of features describing the type of the question Q, where X is a set of features that describe the information bearing features of the answer. For a set of potential answers, the one selected would be the one that maximises the probability of

$$\hat{A} = arg\max P(A|W,X)$$

One of the main observations from this paper was that Log Likelihood was correlated with Mean Reciprocal Rank. Mean Reciprocal Rank as defined in (Bhowan U., McCloskey D., 2015) is the multiplicative inverse of the rank

of the first correct answer. For example for the questions Q_1 and Q_2 ,having correct answers CA_1 and CA_2, if the Question Answering system produces the following list of candidate answers sorted by the weighting (A_1 , A_2, A_3, and A_4 are incorrect answers)

$Q_1 = [A_1, A_2, \mathbf{CA_1}]$

$Q_2 = [A_3, \mathbf{CA_2}, A_4]$

The reciprocal rank of Q_1 is 1/3 and Q_2 is 1/2. For the system, Mean Reciprocal Rank as (1/3 + 1/2)/2 which is about 0.417.

Heie et al (Heie, M., Whittaker, E., Furui, S., 2010) identified that there is a high correlation between Log Likelihood and MRR as shown in Figure 2.3-1

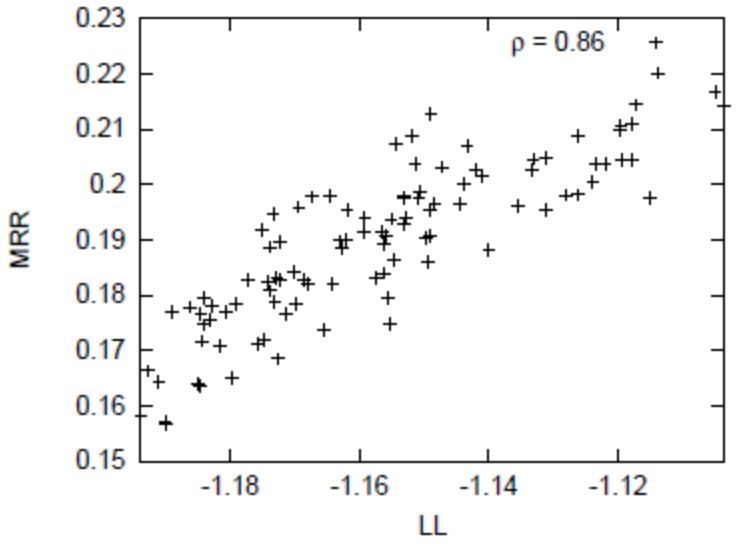


Figure 2.3-1- MRR vs LL (Heie, M., Whittaker, E., Furui, S., 2010)

2.3.3 Topic Signatures for the Document Retrieval task

Topic signatures are sets of related words with their associated weights organised around head topics. Topic signatures play a significant role in Information Retrieval, text summarisation (Hovy, Lin, 1996), and ontology

learning (Agirre, Ansa, Hovy, Martinez, 2000). A formal definition of a topic signature is shown below:

$$t_s = \{(w_1,s_1), \ldots, ((w_i,s_i), \ldots\}$$

Where t_s is the topic (head term) and each w_i is a word associated with the topic, with strength s_i. The strength of each associated word can be assigned automatically using statistical methods based on the frequency of a word in a location. We can describe topic signatures as an extension of the collocation hypothesis or even by quoting John Firth's "you know a word by the company it keeps" (Firth, 1957). As we have seen in the Sentence Extraction/Summarisation section, Term Extraction is another discipline of Natural Language Processing that can be used to derive topic signatures. To define that we can combine two term extraction tasks, one in order to identify domain terms and the second one in order to identify terms collocated to the ones that were picked from the first iteration. An example of a topic signature from the work described in this paper is shown in Figure 2.3-2 for the lexical term "network" with WordNet definition "a system of interconnected electronic components or circuits" (sense #5).

network={(address,230.39),(ip,250.88),(protocol,142.13),(layer,214.99),(ethernet,202.64)}

Figure 2.3-2 – Topic signature example

The figure above shows the head term *network*, which is connected to the terms *address*, *IP*, *protocol*, *layer* and *Ethernet*. All those terms are contained within the domain of computer networking. The strengths

associated with each term are calculated using statistical methods described in chapter 3 of this thesis which focuses on the methodology.

Topic signatures were developed for applications where the background knowledge needed did not require the expensive option of a manually created sense tagged corpus.

The approach taken so far for automatic topic signature acquisition is to collect documents relevant to a domain. Within those documents extract signature terms for some of the concepts in the domain and use the topic signatures generated to process natural language engineering tasks.

In Lin, Hovy (2000) a pre-classified corpus was used, and a set of target concepts were identified for the domain. For each of the identified concepts, terms (including bi- and tri-grams and also stemmed words) were collected from the corpus that were highly correlated with the target concept. The number of terms that are collected for each term were set by using empirical cut-off points depending on the weight of each associated term. The weight of each associated term is calculated by Lin using the log-likelihood measure, which is described in section 2.3.2. To evaluate the system's performance, TREC documents separated into relevant and non-relevant sets according to their TREC relevancy judgment were used. The documents were passed through a POS tagger and the root form of each word was picked using WordNet. Also the frequency of each root word as a unigram, bigram and trigram was collected. The Log Likelihood value is calculated for each term with cut-off weight set to 10.83 and confidence level $\alpha = 0.001$ by looking up an x^2 table. The results that came out of this

evaluation indicated that terms with high Log Likelihood value can be considered as good term candidates for each domain. Bigrams and Trigrams have naturally a decrease in the value, which is expected since they occur less often but on the other hand they are more informative. To further evaluate the summary extraction using topic signatures, the summary that was created was evaluated getting the F-score against human created summaries

The F-score formula used is

$$F = \frac{(1+\beta^2)P\,R}{\beta^2 P + R}$$

Which uses the precision (P) and recall (R) measure and β is the relative importance of P and R.

Another example comes from Bowerman, Oakes, Stamoulos (2008) where the task of extracting topic signatures is broken into smaller tasks in order to eliminate human interaction.

Two information retrieval tools and the measure of mutual information were used to create topic signatures in a method developed by Cuadros et al (Cuadros, Padro, Rigau, 2005). In this method, queries were constructed for all senses of specific words. The different senses were retrieved using WordNet which also provided lists of synonyms, hyponyms and hypernyms. The queries passed to the IR tools (ExRetriever and Infomap) were based on Leacock et al. (1998). An example of a complex query passed through an Information Retrieval system for the term *"network"* would be

(network AND (system#2)) OR* (electronic_network) OR computer_network

Figure 2.3-3 – Query example

This type of query includes both monosemous and polysemous relative terms obtained from WordNet. The retrieved corpus was collected and the topic signatures were extracted depending on the relevance provided by each IR system. This system used a set of Senseval-2 documents in order to evaluate the performance. This task uses simple word overlapping (or weighting) measures. This occurrence evaluation measure simply counts the amount of overlapping words between the topic signatures and the test example. When the weighting evaluation measure is used, the weight of the overlapped words is used. The precision, recall and F1 measure were calculated using ExRetriever and Infomap occurrence and weight overlapping. The monosemous strategy seemed to have the best results regarding Precision and Recall. To evaluate the difference in behaviour between Infomap /ExRetriever and Topic signatures the Kappa statistic was used (Equation 2.3 4).

Equation 2.3-5

$$K = \log \frac{P(A) - p(E)}{1 - p(E)}$$

Finally in (Biryukov, Angheluta, Moens, 2005), documents about well-known people were collected, and clustered depending on the person the document referred to. Afterwards statistical methods (TF.IDF, x^2 and log-likelihood ratio) were applied on the corpus to identify words that co-occur with each person. The topic signatures created were then used to answer

user questions about well-known people knowing that the answer was contained in the documents used for the experiment.

A solution to the problem of comparative extractive document summarisation, which deals with generating a short summary showing the differences in a documented for a specific group of documents can provide us ideas of alternative usage of Topic Signatures. In the work of Wang, Zhu, Li and Gong (2012), they extract the sentences of a domain, e.g. news about Bill Clinton, and check the cosine similarity and other features between sentences in order to see categorise sentenced into different domains. Already having the Topic signatures extracted, this information can be used in order to put potential documents into categories or exclude documents that can be part of a signature term that is not present on the query.

Topic signatures serve as a small knowledge base for a domain. In systems where there is no standard knowledge base or known entities to support logical links between terms, topic signatures provide an efficient alternative. In this section, we described the usage of topic signatures in various systems. The next area we will investigate is the sentence extraction, which sometimes is done with summarisation techniques.

2.3.4 Sentence Extraction / Summarisation

Automatic Summarisation techniques produce a single text of as a compressed version of a set of documents with minimum loss of relevant information (Chali, Joty, Hasan, 2009).This definition is very similar to what QA systems are required to produce. Investigating the area of query based summarisation, we can see that the input and output requirements do

match. For this reason we look into statistical summarisation techniques that have not been used in QA systems. An interesting approach comes from Fisher and Roark (2006) where they compare different statistical metrics in order to provide a query based summary. Their work was inspired from previous work in text summarisation/question answering where each sentence is treated as an element whose weight is seen as an importance rank between the sentence and the query. Sentences ranked above a specific threshold or until they meet an appropriate length are included in the summary. Sentence ranking in text summarisation is a technique used in many systems such as NeATS (Lin, Hovy, 2002) where bigrams and log likelihood are used to extract important sections of the document. Erkan and Radev (2004) used centrality features and an algorithm similar to PageRank in order to extract the most important sentences of documents. MEAD (Radev et al. 2004) is also a multi lingual statistical based summarisation tool which uses position, centroids, sub sequences and keywords in order to extract summaries from documents. Also topic signatures that are described in section 2.3.3 are used to extract summaries with results similar to the best summarisation systems (Biryukov et al., 2005). Finally, graph based algorithms in a multi layered approach have also been successful in document summarisation (Mihalcea, Tarau, 2005).

From the implementations mentioned above, there is strong evidence that statistical approaches work very well in the summarisation task. Breaking the Question Answering task in multiple layers, where the final layer will extract the important sentences of the document that contains the answer,

will take advantage of state of the art algorithms and use them in a domain where they have not been used in existing research.

These approaches would also benefit e-Learning systems where supervised ranking approaches would decrease the usability of the application because the responsiveness would not be real time, and also the need of a domain expert for the ranking process will not bring any benefits to the academic staff. In the system presented in 2005 by Maria Biryukov, Roxana Angheluta and Marie-Francine Moens (Biryukov et al., 2005) system, there are three main areas:

- Normalise and segment sentences
- Rank sentences either focused on a query or not
- Select the appropriate sentences from a ranked list

For each sentence in a cluster of documents some word based statistical features (TF*IDF, log likelihood and log odds) are calculated. The equations for the metrics are described in chapter 2.3.2.

Additional features for each sentence would be the sum and average score of each sentence. An improvement to the algorithm (Biryukov et al., 2005) came by using the neighbour sentences features added to the sentence under investigation. The result of this improvement would allow to locate collections of sentence as “hot points” of the text that have similarity with the query and also the presence of query terms in the neighbouring sentences would make the text more important.

The evaluation of the system was made using the ROUGE package (Lin, 2004), which basically compares a summary with another ideal summary

created by humans. There are a few measures in ROUGE (ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S etc.) mainly used by the Document Understanding conference (DUC). Features that are considered by ROUGE are the overlap of n-grams, word sequences, pairs etc.

The work of Gelbukh et al. (2010) used log likelihood in order to extract keywords from a collection of documents. Although the area of Summarisation may seem a bit distant from Question Answering, there is a similarity in trying to identify important terms in document that will return the correct answer from a collection of documents, with using the important terms to return a summary of the document. The assumption behind that is that the summary of the document that contains the correct answer will also contain the correct answer. An important quote guiding our work comes from John Firth - "you know a word by the company it keeps" (Firth, 1957) and tells us to look into collocation of words. So basically, if a word is important in a document, which can be picked up using term extraction techniques and the collocations match the query, there should be a good chance that the answer to a question would be included in the text.

To look a bit further into the work of Gelbukh (Gelbukh et al. 2010), the main aim of the work was to initially extract single word terms for a specific domain and then to use the terms extracted in order to identify multi word terms. In order to identify the terms, log likelihood was used because it performs better than traditional methods such as TF*IDF (He, Zhang, Xinghuo, 2006). A reference corpus was used as well in order to get some baseline metrics on word frequencies. The corpus was a large collection of

general documents. The size of the reference corpus is quite important, since we need to have a large corpus in order to pick differences between the frequencies corresponding to a term or pair of terms. Also the pre-processing of the corpus does not use any external knowledge bases that provide enrichment such as WordNet. The log likelihood formula used is described in section 2.3.2. An important aspect in the method was that if the relative frequency of the term in the collection was not greater than the frequency of the term in the reference corpus then for evaluation of the results they performed an experiment of extraction and manually scored the responses. Afterwards the precision and recall measures were used. The formulas used to calculate precision and recall are presented in section 2.4

Other statistical models have also been used for sentence ranking. Relative Entropy is one (Kumar C., Pingali P., Varma V., 2009) which is th KL-Divergence of the sentence model M_S with the document M_D. The entropy is calculated using Equation 2.3-6

Equation 2.3-6

$$S_{KL} = D_{KL}(M_S \,||M_D) = \sum_w P(w|M_S) \log \frac{P(w|M_S)}{P(w|M_D)}$$

And sentence relevance is defined as the reciprocal of S_{KL}.

An interesting approach in summarisation comes from You Ouyang et al (2013) where the feature for sentence relationship is explored. According to their work, sentence relationship is the recommendation degree of a sentence by another. For example if sentence A is selected for a summary,

we check how much sentence B needs to be included in order to support the concepts in summary A. This method works on the hypothesis that a single word is not sufficient to represent complex contexts and also sometime ambiguity of terms can introduce errors to the summary.

The multi feature approach that has been used in document retrieval has also been used in summarisation. Yogesh Kumar Meena and Dr. Dinesh Gopalani (2014) researched in different features that have been used across the year to create summaries. These features included TF.IDF, word co-occurrence, named entities, sentence location etc. The combination that scored higher combined features of TF.IDS, sentence similarity and sentence location.

The multi feature approach is also adapted in the paper “Applying regression models to query-focused multi-document summarization” by Ouyang et al (2011). In this paper they investigated on features for query based summaries. The statistical features used are the word matching feature, where sentences that include query terms rank higher than ones that do not, Word TF.IDF, to scale in information richness of a word. The linguistic features that are used, are the named entity feature, and named entity matching feature and semantic matching. Finally some morphological features are used, such as Stop Word Penalty and Position.

Gabriel Silva et al (2015), evaluated 20 featured falling into three main categories in their work. The main categories were word based ones, where most important words are scored, sentence based features, where features such as the position of the sentence, similarity to the title etc. and

finally graphic, that used the relationships between words and sentences. From the selected features, language independent ones were preferred to be used in order to allow multi language summarisation. The overall accuracy of the summary was 52% on the unbalanced basis and70% on the balanced basis.

The multi feature approach is also adopted to more linguistic approaches as well. A combination of TextRank (Mihalcea R., Tarau P., 2004) with similarity metrics from WordNet and the position of the sentence is the document is used in the work of Araly Barrera and Rakesh Verma (Barrera A., Rakesh Verma R., 2011)

2.4 Evaluation metrics

Once the algorithm is in place, there is the need for the system to be evaluated. The evaluation of the Question Answering system according to Ravichandran and Hovy (2002) is dependent on two different components, the accuracy of the Document Retrieval and of the Answer Pinpointing modules. A good evaluation metric for the Document Retrieval part is recall and precision. The precision metric is calculated using Equation 2.4-1 :

Equation 2.4-1 - Precision

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Where the recall metric can be calculated using the formula below:

Equation 2.4-2 - Recall

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

There are many published variations of these metrics, by modifying their attributes. For example, for a large document collection, the top N documents can be taken into consideration when calculating the metric. Also in the Question Answering domain, one can input to the system under evaluation a set of questions that their answer is known and measure against correctly answered questions, or correctly identified top documents.

Recall can also be used to evaluate the answer pinpointing module. The way to implement that would be to use relevant sentences and retrieved sentences as the inputs to the formula. This can identify how well the algorithm operates, by picking only relevant sentences out of the documents identified by the Document Retrieval algorithm.

In early tests of retrieval systems (pre 1994), there were some empirical findings that there is a trade-off between Precision and Recall. This triggered of a research by Michael Buckland and Fredric Gey (Buckland M., Gey F., 1994) to investigate into a mathematical model on the trade-off between the two metrics. Their findings indicated that the trade-off is not only an empirical finding, but there is a mathematical explanation behind it.

Davis and Goadrich (2006) have proven that there is a trade-off between precision and recall. The trade-off occurs when the number of documents retrieved increases and the retrieval performance in equal or less to the value before the retrieved documents increased. To formalise that, they identified that if Recall is modelled by a polynomial function of proportion of documents retrieved, then Precision is modelled by a lower order polynomial of the same value, as shown in the Figure 2.4-1 .

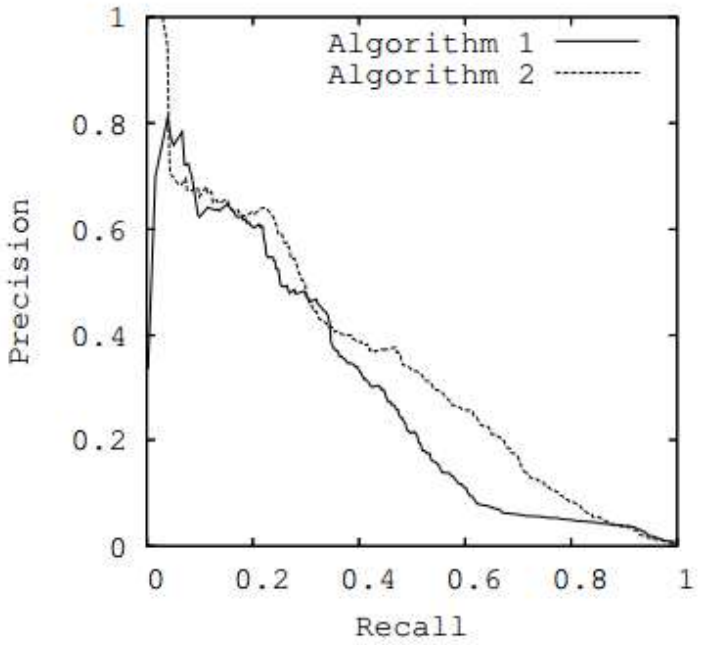


Figure 2.4-1 –Precision over Recall (Davis, Goadrich, 2006)

To overcome this barrier and in order to provide a more accurate score for an answer or any other measurable outcome another metric that combines precision and recall is widely used which is the F_β family.

Equation 2.4-3 – F-score formula

$$F_\beta = \frac{(1+\beta^2)p \cdot r}{\beta^2 \cdot (p+r)}$$

Where β is a parameter that controls the relative importance of recall and precision.

Having β set to 1, the F_1 measure is the harmonic mean of precision and recall which is closer to the minimum value of the combination of the two metrics as opposed to a mean value which is closer to the maximum value of the combination of precision and recall.

Precision is a metric widely used in information retrieval, where the top n documents are evaluated for precision. In Question answering systems, the Precision@1 (precision of the top document) is the metric that the system needs to be optimised to (Agarwal, A., Raghavan H., Subbian, K., Melville, P., Lawrence, R., C. Gondek, D, Fan, J., 2012)

2.5 Literature summary

In this chapter we looked into a variety of Question Answering in order to investigate the current state of research. The first outcome was to come up with a generic multi module approach to use for the Question Answering system. Using both statistical and Language based systems we extracted a more generic framework to base our system that contained 3 main modules, Query Parsing, Document Retrieval and Answer Pinpointing.

For each of these modules, we looked into techniques within the Question Answering research but also in other areas of Computational Linguistics. In more recent research in QA and Summarisation, the trend is to use a multi feature approach which is something our system is based on. In the Query Parsing module, apart from using standard approaches on identify the most important terms, Local Analysis techniques (Xu, Croft, 1996) are used in order to assign a more specific importance score at each term. Log Likelihood is the preferred statistical score since it produced good results in (Soricut R., Brill E., 2006) for co-location retrieval and also to extract keywords(Gelbukh et al. 2010). We took the query processing a step further by assigning a score to each term based on their IDF score in order to specify the importance of each term.

On the document retrieval task, we experimented with different features used again in both statistical, hybrid and language based systems. For example, the scoring of documents higher that contain more keywords than others (Ouyang et al., 2011) and also TF.IDF as a metric. From the linguistic approaches we saw an improvement on retrieval due to the use of background knowledge. Since statistical approaches are not backed up with knowledge bases, we used statistical approaches in order to create a dynamic knowledge base based on Topic Signatures research. After extracting the topic signatures, we used them in a way of classifying the documents in a domain depending on their signature terms and also exclude documents that are not part of the domain specified by any signature terms in the query.

Finally for the Answer Pinpointing task, we looked into multi feature systems (Barrera A., Rakesh Verma R., 2011), (Gabriel Silva et al (2015), and also (Meena K., Gopalani D., 2014), (Ouyang et al., 2011). We then looked at the most applicable morphological feature in order to come up with an algorithm to minimise the number of sentences returned by the system.

Chapter 3

3 Methodology

3.1 Introduction

This chapter covers the approach taken to develop the system that provides automatically generated help in response to a student's query. In this chapter, the process of implementing a system in order to prove our hypotheses is described in detail.

Figure 3.1-1 shows the main parts of a Question Answering system as they can be identified from the relevant literature. In existing systems, some of the components may be broken down into more modules, but in figure 3.1-1 we have a higher view of the architecture. The architecture consists of a Query parsing module which will identify the main terms of the question the user enters into the system. The terms are then passed to the Document Retrieval module which identifies the document that contains a candidate answer to the question. The candidate document is then analysed by the Answer Pinpointing module with the use of the keywords picked up by the Query Parsing module.

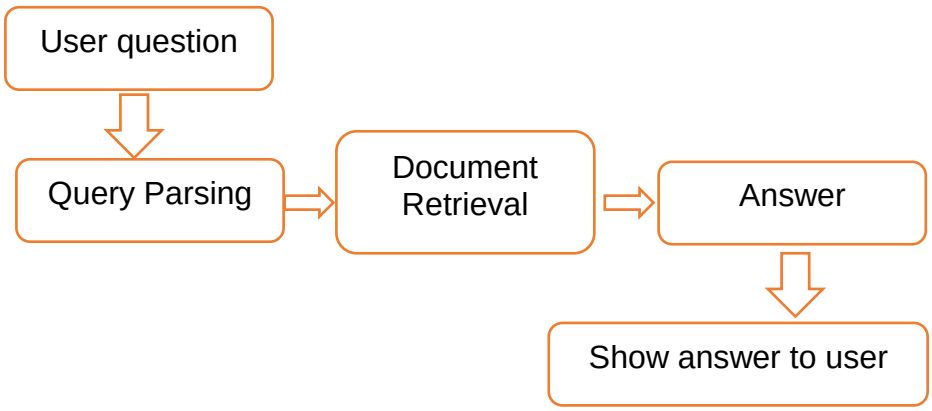


Figure 3.1-1 – QA System overview

The rest of the chapter is organised in sections corresponding to each module as shown in figure 3.1-1. Each section describes the different

development phases explaining the aim of the developments, the results and the need for a next phase if required. We start the next section with a set of objectives that need to be met in order to evaluate our hypotheses and answer the research questions.

3.2 Objectives

Our main objective is to investigate statistical approaches that can be used to retrieve answers to user questions. In order to accomplish that we will need to meet the following objectives:

Table 3.2-1 – Main objectives

Objective 1	Identify the important words from the user question	Query Parsing
Objective 2	Weigh the important words according to importance	Query Parsing
Objective 3	Identify documents that contain the keywords picked up from objectives 1 and 2	Document Retrieval
Objective 4	Weight the importance of the documents identified	Document Retrieval
Objective 5	Pick the document containing the answer using only statistical approaches	Document Retrieval
Objective 6	Create an answer using sentences of the document	Answer Pinpointing

Table 3.2-2 provides a map of the objectives in relationship to the research questions and hypotheses.

Table 3.2-2 Questions – Objectives – Hypotheses Map

Questions	Objectives	Hypotheses
Can a QA system using statistical based techniques provide the similar level of answers as a baseline search engine?	Objective 1 - Identify the important words from the user question Objective 2 - Weigh the important words according to importance Objective 3 - Identify documents that contain the keywords picked up from objectives 1 and 2 Objective 4 -Weight the importance of the documents identified	(H1) The correct answer to a question entered to the system should be retrieved using statistical methods and without requiring any background knowledge. (H3) A good combination of methods that will work on a learning domain to answer user specific questions are: **Log Likelihood** – to measure the importance of query terms and assign term and document weights **Summarisation techniques** - to extract sentences relevant to the user query
How well will our algorithm perform using a smaller or a larger corpus since the amount of documents in the VLE will be different per institution?	Objective 4 -Weight the importance of the documents identified Objective 5 - Pick the document containing the answer using only statistical approaches	(H1) The correct answer to a question entered to the system should be retrieved using statistical methods and without requiring any background knowledge. (H2) The statistical approaches used will not be dependent on the size of the corpus and the system should be able to retrieve the correct answer having a small or a large corpus to use for weight calculations.
Is there a categorisation technology such as topic signatures that can be improved from the current state and used within our algorithm in order to support students?	Objective 5 - Pick the document containing the answer using only statistical approaches	(H5) Topic signatures can be acquired and used for computational tasks, using local analysis techniques and statistical weights without the intervention of an expert user.
Summarisation is an Information Retrieval technique that, given a document, returns the important sentences of a document. Would that kind of technique be of use for choosing the answer to a user question?	Objective 6 - Create an answer using sentences of the document	(H3) A good combination of methods that will work on a learning domain to answer user specific questions are: **Log Likelihood** – to measure the importance of query terms and assign term and document weights **Summarisation techniques** - to extract sentences relevant to the user query

The next section describes how the Query parsing module was developed and enhanced in order to achieve the objectives 1 and 2 described in table 3.2-1

3.3 Procedures

For the hypotheses to be tested some formal procedures needed to be followed.

P1: Retrieve the correct answer:

I. Provide each of the CCNA questions as input to the system.
II. Retrieve the answer from the system using the CCNA corpus to pick an answer by using only the CCNA corpus for any statistical calculations.
III. Compare the answer with the one provided by CISCO. If all the content of the self-assessment question is covered then state the outcome as success

P2: Ensure performance does not drop when the corpus size grows

I. Use the CCNA questions as input to the system.
II. Retrieve the answer from the system using the CCNA corpus to pick an answer, and the CCNA corpus and the reference corpus for any statistical calculations.
III. Compare the answer with the one provided by CISCO. If all the content of the self-assessment question is covered then state the outcome as success
IV. Compare the amount of correct questions of this experiment with the amount of correct answers of P1. There should not be any drop in the performance of the system

P3: Investigate the most appropriate techniques

I. Use the CCNA questions as input to the system.
II. Retrieve the answer from the system using the CCNA corpus to pick an answer and the CCNA corpus and the reference corpus for any statistical calculations. The statistical calculations would be done using Log Likelihood and TF.IDF and also using sum and average

of the measures of each term extracted by the Query Parsing module.

III. Successful outcomes for each statistical measure are the ones that contain the full answer that is provided by CCNA.

IV. Compare the measures and identify the one with the most successful outcomes

P4: Acquire topic signatures

I. Use the CCNA questions as input to the system.

II. Use statistical methods to identify topics in query term.

III. From the documents that contain the topics, measure the statistical weight of other terms and how often they co-appear in the same document.

IV. Investigate a threshold value that can be used to separate if a term is a topic term or a signature term.

V. Having a set of topic terms identified from the previous step, use these terms to retrieve signature terms. To acquire signature terms, calculate the Log Likelihood of the term in the set of documents that contain the topic term from the CCNA corpus. Normalise the log likelihood using IDF. Keep as signature terms the terms that are outside the 95% of the distribution.

P5: Provide answers quicker to students using less steps

I. Use the CCNA corpus and a full text search interface like Lucene to search though the documents.

II. For this procedure we need a set of students with basic understanding of computing, willing to answer a set of question.

III. A web system is required to allow students to:
    a. See the questions;
    b. Search through the CCNA corpus to find an appropriate answer;
    c. Select the document that they think the answer is located in;
    d. Collect the number of searches per question;

e. Collect the time spent on each question;
f. Collect the documents the student opened in order to find the answer

IV. Use the CCNA self-assessment questions and ask the student to find the document that they think the correct question is included in using the search engine.

V. Collect all the data stated in step III

VI. Display the answer provided by the Question Answering system and prompt each students to select which answer he/she prefers, between the answer manually retrieve by him/her against the answer generated by the Question Answering System.

3.4 Query Parsing

3.4.1 Aim

As mentioned before, the Query Parsing module will take as input the user query and extract the keywords and also assign weights for each of the keyword terms. The weights will be used in order to provide the other modules more details about the terms. This module aims to accept a question as input and return a list of important terms with their associated weights. What the evaluation will consist of in this part of the system is if the words identified as key terms are actual key terms and if the words that are identified as less important do not carry any significant information about the query and also if the expansion techniques we use are adding any value to the retrieval of the correct document.

3.4.2 Phase 1 – Pilot Run

For the Query Parsing module, five different runs were conducted and the results were recorded for evaluation. These phases started from a baseline system that identifies keywords based on the criterion of the term not being

present in a stop word list, and expands to bigram identification, assigning weights to each keyword and also automatically extracting topic signatures based on the query terms

3.4.2.1 Aim

The aim of this phase is to identify how far simple approaches such as term filtering can benefit the Query Parsing module. The phase in this section describes the use of a stop word list to identify important query terms.

3.4.2.2 Implementation

On the first run of the application a stop word list was used in order to remove any stop words from the query. The stop word list is described in section 1.6.1. Stop words are commonly used words in a language that carry no special meaning and can be ignored.

In this run we individually run only the Query Parsing module using the stop word list to filter out unimportant terms. Our aims targeted creating a list of important terms for each of the questions selected to conduct the evaluation (Table 4.2-1 – Self Assessment Questions). The stop word list described in 1.6.1 is solely used in this experiment. Using the stop word list creates a language specific dependency, since the stop word list is developed for the English language only. To evaluate the first run of our algorithm the output list is checked to identify if any of the removed terms were carrying any meaning related to the query and also if any of the terms included in the query carry no meaning for the query. A sample of the results of this run is shown in the next section 3.4.2.3. The full results of this initial run are shown in section 4.2.1.1.

3.4.2.3 Results

The results for the phase 1 run are shown in Table 3.4-1

Table 3.4-1 - Query Parsing phase 1 results

Question	Retrieved terms	Non-stop words extracted	Non-stop words in question	Non-stop word precision (%)	Non-stop word recall (%)
Describe the use of a network interface card (NIC)?	network interface card NIC use describe	6	6	100	100
Describe the rated throughput capacity of a given network medium?	network capacity throughput medium given rated describe	7	7	100	100
What describes a LAN?	LAN describes	2	2	100	100
Why was the OSI model created?	created OSI model	3	3	100	100
Why are the pairs of wires twisted together in an UTP cable?	cable wires UTP twisted pairs	5	5	100	100
What is required for electrons to flow?	required flow electrons	3	3	100	100
How does using a hub or a repeater affects the size of the collision domain?	does using size repeater Hub domain affects collision	8	7	87.5	100
What will cause a collision on an Ethernet network?	network Ethernet cause collision	4	4	100	100
What Ethernet implementations use rj-45 connectors	Use Ethernet implementations connectors	4	5	100	80
What are the functions of a router in a network	network router functions	3	3	100	100

The first issue with this approach is that there is a loss of information by treating multi-term words as single terms. The list of terms will be fed into the Document Retrieval module, which will then look for the document that contains the terms. Question in row 1 is referring to *network interface cards*,

whereas the second row is about *network mediums*. Both contain the term *network* in their term list which will return more results.

3.4.2.4 Need for next phase

The need for a next development phase arises from the fact that the terms extracted may be a part of a domain bigram. Bigrams are two words usually found together in the corpus that have a more specific meaning than the two words individually. By not using bigrams, irrelevant documents will be retrieved by the Document Retrieval module. This can skew the statistical weights which in the subsequent modules can cause the incorrect answer to be retrieved.

In the next section, the approach taken to retrieve bigrams form the CCNA corpus is described, so they can support the Query parsing module.

3.4.3 Phase 2 - Bigram Identification

3.4.3.1 Aim

The aim of this phase is to detect any benefits a bigram identification enhancement will provide to the overall Question Answering system. From the previous run, some irrelevant terms are filtered out and will not be passed to the Document retrieval module, and the aim of this phase is to identify any bigrams that are present in the question and treat them as a single term. Bigrams are two word terms that are often used together. From a statistical perspective, bigrams hold more information than single terms because they will be used less often that the terms alone in general language

3.4.3.2 Implementation

For the implementation of this phase the questions used in the evaluation are the ones in Table 4.2-1 – Self Assessment Questions. For the statistical weights, the term frequencies in the CCNA corpus derived as described in section 1.6 are used.

Collocation related information corresponding to the terms is collected from the Cisco CCNA corpus and used by the Query Parsing algorithm. For example if a bigram “network cable” is present in the query, documents that contain the bigram will be more relevant to the query than documents that contain single occurrences of the terms that compose the bigram (“network” or “cable”). Not using the information we can get from bigrams can return the wrong results. So the first addition we add to the system is the bigram identifier.

A bigram should be treated as a relatively more important term since if a potential document contains query bigrams, there is a greater probability for the correct answer to be in the sentences of that document. With a stricter document selection algorithm as described in phases 1 and 2 of the Document Retrieval module (section 3.5), documents that do not contain the bigram will not be picked by the document retrieval module. Specifically, if two words were identified as bigrams in the query, they are used as such in any statistical calculations. Using the example in the previous paragraph, this part of the algorithm will filter documents that mention the term *network* and not have the second part of the bigram which is *cable*. The bigram list will be maintained automatically from the corpus data available in the system.

The weight of each bigram will be assigned using the log likelihood statistical measure which is described in section 2.3.2. To explain how the weight of each bigram is calculated the bigram *network cable* is going to be used, where the word *network* is w_1 and *cable* is w_2.

Initially four different frequencies are then calculated which are:

1. The occurrences of w_1w_2 ($\overbrace{netwok}^{w1}$ $\overbrace{cable}^{w2}$)
2. The occurrences of $w_1\overline{w_2}$ ($\overbrace{network}^{w1}$ $\overbrace{any\ word\ except\ cable}^{\overline{w2}}$)
3. The occurrences of $\overline{w_1}w_2$ ($\overbrace{any\ word\ except\ network}^{\overline{w1}}$ $\overbrace{cable}^{w2}$)
4. The occurrences of $\overline{w_1}\ \overline{w_2}$ ($\overbrace{any\ word\ except\ network}^{\overline{w1}}$ $\overbrace{any\ word\ except\ cable}^{\overline{w2}}$)

These frequencies will be plotted in a table for each potential bigram, shown in the figure in Table 3.4-3as used in (Baroni, Evert, 2011).

Table 3.4-2 – Bigram log likelihood

w_1w_2	$w_1\overline{w_2}$
$\overline{w_1}w_2$	$\overline{w_1}\ \overline{w_2}$

For readability, we name the observed frequencies of the table above as

$O_{11} = w_1w_2$

$O_{12} = w_1\overline{w_2}$

$O_{21} = \overline{w_1}w_2$

$O_{22} = \overline{w_1}\ \overline{w_2}$

So the table can be written using the observed frequencies as

Table 3.4-3 – Log likelihood observed frequencies

First bigram term	Second bigram term		Row sums
	w_2	$\overline{w_2}$	
w_1	O_{11}	O_{12}	R_1
$\overline{w_1}$	O_{21}	O_{22}	R_2
Column Sums	C_1	C_2	$N=(R_1+R_2+C_1+C_2)$

The row sums (R_1, R_2) and the column sums (C_1,C_2) are also calculated, with the total number of occurrences being N.

Another calculation needed is the *expected frequency.* The *expected frequency* for each pair of terms is shown in Table 3.4-4.

Table 3.4-4 – Bigram Estimate frequencies

First bigram term	Second bigram term	
	w_2	$\overline{w_2}$
w_1	$E_{11}=\frac{R_1C_1}{N}$	$E_{12}=\frac{R_1C_2}{N}$
$\overline{w_1}$	$E_{21}=\frac{R_2C_1}{N}$	$E_{22}=\frac{R_2C_2}{N}$

Having all the observed and expected frequencies calculated, the log likelihood of the term can be calculated using the Equation 3.4-1

Equation 3.4-1 – Log Likelihood Formula

$$LLR=2*\left(\left(O_{11}*\log\left(\frac{O_{11}}{E_{11}}\right)\right)+\left(O_{12}*\log\left(\frac{O_{12}}{E_{12}}\right)\right)+\left(O_{21}*\log\left(\frac{O_{21}}{E_{21}}\right)\right)+\left(O_{22}*\log\left(\frac{O_{22}}{E_{22}}\right)\right)\right)$$

To implement the bigram retrieval, another step is added to the data preparation (1.7). This step uses all possible term pairs in the documents

and calculates and stores them in the database described in section 3.8. This makes the identification of Bigrams faster, since all potential bigrams are already stored in our database with their observed frequencies. When two terms are checked to see if they are a bigram, there is no need to look into the corpus, just a check in the database will retrieve the data required. To limit the amount of data stored, if any of the bigram terms is a stop word which can be found in the stop word list, the bigram is discarded. An example of the bigrams retrieved is shown in Table 3.4-5.

Table 3.4-5 – Bigrams identified

Term1	**Term2**	**Frequency**
Web	Links	171
TCP	IP	144
IP	address	127
Interactive	media	105
media	Activity	102
IP	addresses	81

Having the Bigram frequencies stored, the next step of the algorithm is to calculate the estimated frequencies since the bigram frequency can be used as the observed frequency and everything else can also be calculated from the existing stored records.

In the frequency column we can see the value of w_1w_2 of the reference table we use. It is fairly easy to calculate the other frequencies needed for the reference table by adopting queries to count specific sums.

For the Bigram "**Web Links"** we have the following values where w_1 is Web and w_2 is Links

Table 3.4-6 – "Web Links" observed frequencies

	w_2	$\overline{w_2}$	
w_1	171	48	$R_1 = 219$
$\overline{w_1}$	27	21,896	$R_2 = 21923$
	$C_1 = 198$	$C_2 = 21944$	N=22,142

We can also calculate the estimated frequencies for the bigram

Table 3.4-7 - "Web Links" estimated frequencies

	w_2	$\overline{w_2}$
w_1	$E_{11} = \frac{R_1 C_1}{N} = 2.166$	$E_{12} = \frac{R_1 C_2}{N} = 199.061$
$\overline{w_1}$	$E_{21} = \frac{R_2 C_1}{N} = 196.014$	$E_{22} = \frac{R_2 C_2}{N} = 21726.958$

Equation 3.4-1 is then used in order to assign a weight to the bigram *Web Link*.

The final step of the algorithm is to identify which bigrams are going to be significant and which are not. To do that we rely on the chi squared distribution with one degree of freedom in order to determine the statistical significance of the log likelihood radio. The significance value tells you how often a calculated Log Likelihood ratio can occur by chance. So from the Table 3.4-8, a weight of 6.63 can occur by chance only about one in a hundred times, so the significance is 0.01.This significance measure will be

applied to all bigrams in the Cisco CCNA corpus that have a Log Likelihood Ratio (LLR) over 6.63

Table 3.4-8 – Log likelihood ratio significance.

LLR	Significance
15.13	$p < 0.0001$
10.83	$p < 0.001$
6.63	$p < 0.01$
3.84	$p < 0.05$

Significance of 0.01 is used as a cut off point for significant and insignificant bigrams. So if a bigram has a weight below 6.63, it means that it will only appear by chance, so it is not important, one in a hundred times. All the other occurrences will be treated as significant. In section 3.4.3.3 a sample of the results near the cut-off point is shown and also bigrams identified on the higher end of the log likelihood.

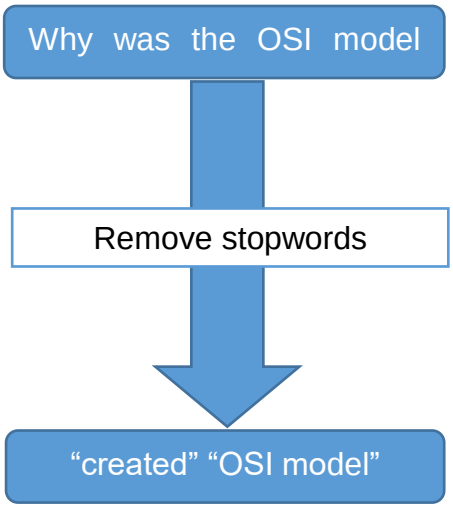
Bigrams were used in such a way in the Question Answering system so as to adjust a sentence weight when being passed to the Document Retrieval module. When a bigram is present in the query it is then picked up by the Question Parsing module. Then the Document Retrieval module instead of using the bigram as two separate terms, it just picks documents containing the bigram.

Local analysis techniques which were described in section 2.3.1.1 are used to expand and classify the queries entered by the users. This is in order to provide better potential answers to the Answer Pinpointing module.

To explain the need for the algorithm of parsing a question we will use a relatively small question that did not score well when we used the QA system with no expansions. The question we will use is “Why was the OSI model created?”

Using the algorithm described above, the following steps would occur when parsing the question “Why was the OSI model created?” through this phase of the algorithm.

1. Examine in order to determine which of the terms a potential bigram is.
2. Check if “created OSI” is a bigram, which returns no results in our database
3. Check is “OSI model” is a bigram, which returns a result with the weight of 259.92 which means is a significant one.
4. The keywords that are passed to the Document Retrieval module are ‘created’ and ‘OSI model’



When looking to understand the query, the document selection that is used is important. Local techniques are used for this reason, as described in section 2.2.1.1. Local techniques provide good information about query terms, by creating a sub corpus of all relevant documents. So initially a list of documents that contain the keywords from the query are retrieved.

Using *OSI model* as a bigram as identified by the weight in the corpus, we get 60 distinct documents returned to extract the answer from. Using *OSI model* as 2 different terms (*OSI* AND *model*), so our keywords are **OSI**, **model** and **created,** the potential documents to pick an answer from are 90 documents. Having the document that contains the correct answer in both sets, the set using the bigram has a better recall score and is preferred than using the *OSI model* as 2 different terms.

3.4.3.3 Results

Some bigrams at the lower end of log likelihood ratio score are shown below. Table 3.4-9 shows bigrams with weight near 6.63 that were correctly identified as bigrams. The results show then the majority of the bigrams in that area are not very relevant to the domain of the corpus. Similarly the bigrams under the threshold point are not important for the corpus.

Table 3.4-10 shows part of the adjacent terms that appear around the significance threshold within the corpus but are not bigrams. The closer to the threshold the potential bigram is, the greater the possibility not to be used by the domain or not being a bigram.

The final set of results in this section come from the higher end of weights.

Table 3.4-11 displays top identified bigrams.

Table 3.4-9 -Bigrams near selection threshold

term 1	term 2	frequency	weight
session	layer	3	6.66181
longer	distance	2	6.65128
addressed	interfaces	1	6.64836
automatically	configuring	1	6.64836
automatically	negotiate	1	6.64836
Capacitor	electronic	1	6.64836
collect	Mail	1	6.64836
complicated	task	1	6.64836
easily	monitored	1	6.64836
intermediate	splitters	1	6.64836
memory	RAM	1	6.64836
memory	ROM	1	6.64836
passwords	user	1	6.64836
sensitive	electronic	1	6.64836
SMTP	administers	1	6.64836
zip	code	1	6.64836
shared	radio	2	6.63738
large	LAN	3	6.63297

Table 3.4-10 - Bigrams below selection threshold

term 1	term 2	frequency	weight
10x10x10	1000	1	6.64836
128	respectively	1	6.64836
2346	bytes	1	6.64836
rejection	characteristics	1	6.64836
sample	32	1	6.64836
seen	outside	1	6.64836
microscopicsized	electronic	1	6.64836
nodes	ad	1	6.64836
nodes	attempting	1	6.64836
nodes	wait	1	6.64836
contain	dozens	1	6.64836
credibility	just	1	6.64836
D	2000	1	6.64836

Table 3.4-11 – Top Bigrams

term 1	term 2	frequency	weight
TCP	IP	144	521.9349
Interactive	media	105	439.7387
media	Activity	102	348.4864
gigabit	Ethernet	69	268.4108
IP	address	127	265.8489
OSI	model	70	259.9208
layer	2	78	209.3932
Optical	Fiber	58	205.5996
IP	addresses	81	177.7789
transport	layer	55	174.3558
Mac	address	62	159.5313
collision	domains	41	154.1835

An interesting point is that because of some structural points of the documents such as a *web link* hyperlink at the end of each page, the algorithm returns such elements as bigrams. For example the following terms in Table 3.4-12.

Table 3.4-12 – Erroneous bigrams

term 1	term 2	frequency	weight
Web	Links	171	701.8527
Lab	Exercise	48	217.2208
Activity	Lab	48	212.2887

Since the bigrams will only be used as a portable knowledge base, the above should not create a problem to our system since the terms above have a small probability to be used for a student question on their own. By introducing more terms, the most relevant document will still score higher and the contrary in the situation where a proper bigram is not used as bigram in the Document Retrieval stage of the Question Answering System.

Also in the top 45 identified bigrams there was only one pair of terms that was not a bigram (0 0), which is a very positive result.

3.4.3.4 Need for next phase

The addition of the bigram retrieval stage added substantial information to the query terms. There is still a chance that the question will not have any bigrams and all the processing would be done by the algorithm developed in phase one which does not include any extra information about the terms. Another enhancement added to our existing Bigram extraction will try and assign “importance” weights to the terms. The development for this enhancement is described in the next section.

3.4.4 Phase 3 – Term Weights

3.4.4.1 Implementation

One piece of the information our terms require to have is the importance of each term in the question. Having all terms as equal in the question brings a limitation to the system that is not true in everyday usage of language. If a question is asked to a person, there would be some important terms that will define the answer, but out of the important terms of the question, some are more important than others. Not all terms are of equal importance in a question and a statistical approach can be used in order to assign weights to each term. The way the algorithm works to assign weights is to use a technique derived from what would be considered local analysis.

For each non-stop word term, single word or bigram, the documents from the CCNA corpus are retrieved. The IDF weight for each term is calculated using the formula in Equation 3.4-2

Equation 3.4-2 – Inverse Document Frequency formula

$$IDF(t) = \ln\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ the\ term\ in}\right)$$

The outcome for each term in the query is a map of the term and the associated IDF frequency for each question. For example for the question *"Why was the OSI model created?"* the flow is shown in Figure 3.4-1

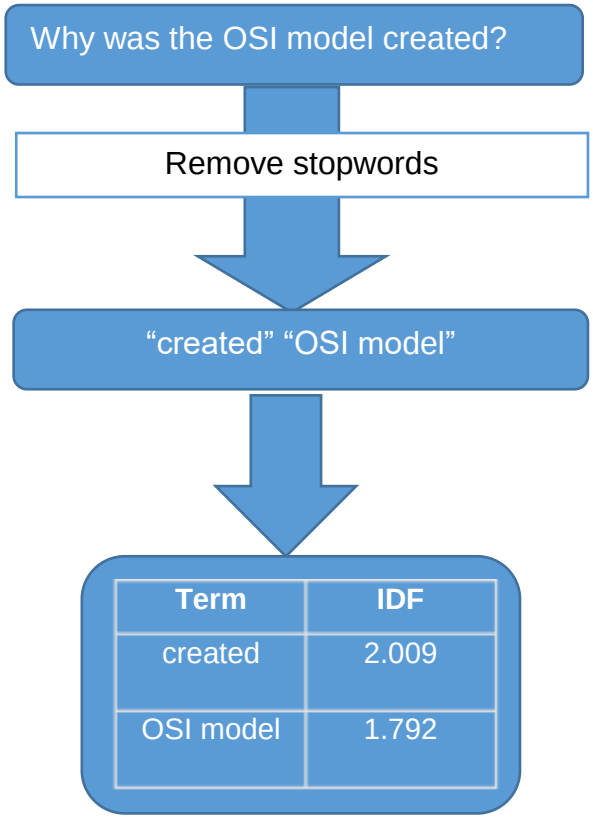


Figure 3.4-1 – Flow of Question Parsing module

The next step is to normalise the weights. The example query is based on 2 terms so assuming that the sum of the weights of the two terms should be 1, the ratio of each term is then calculated. The sum IDF for a query having *p* terms is shown in Equation 3.4-3

Equation 3.4-3 – Sum of IDF formula

$$IDF_{sum} = \sum_{n=1}^{p} IDF_n$$

In our case *p=2* so

$$IDF_{sum} = IDF_{created} + IDF_{OSI\ model} = 7.45 + 3.51 = 10.96$$

The normalised IDF for each term is shown in the Equation 3.4-4

Equation 3.4-4 –Normalised IDF

$$term\_weight_t = \frac{IDF_t}{IDF_{sum}}$$

So for the terms above, we have the following weights used in the formula in Equation 3.4-4.

Equation 3.4-5

$$term\_weight_{created} = \frac{IDF_{created}}{IDF_{sum}} = 0.68$$

Equation 3.4-6

$$term\_weight_{OSI\ model} = \frac{IDF_{OSI\ model}}{IDF_{sum}} = 0.32$$

From Equation 3.4-5 and Equation 3.4-6 it is shown that the score was not as we expected, since the term *created* scores higher than the bigram *OSI model*. The answer to this discrepancy is that the term *create* appears in many forms in the corpus each having different frequencies as shown in Table 3.4-13

Table 3.4-13 – Terms without stemming

Term	CCNA notes Frequency	IDF
create	47	7.028571
created	33	5.72093
creating	5	49.2

To correct this issue the stem of the term can be used the stem of the term when calculating the frequencies. By stemming a term, only the root of the term is used so words like create, creating and created would have the same stem.

Stemming yields the stats as shown in Table 3.4-14

Table 3.4-14 – Stemmed weights

Term	Weight
OSI model	0.56
create	0.44

Which illustrates that the OSI model term is more important in the query when the terms identified as important are stemmed before the IDF is calculated.

3.4.4.2 Results

The full results of this evaluation are shown in section 4.2.1.3, but to demonstrate the enhancement obtained using stemming, the first two questions are passed to the Query Parsing module and the weights of each of the terms are shown in Table 3.4-15

Table 3.4-15 – Terms and bigram weights with and without stemming

	Using stemming		Not using stemming	
Question	**Term**	**%weight**	**Term**	**%weight**
Describe the use of a network interface card (NIC)?	interface card	0.368	interface card	0.311
	network interface	0.324	network interface	0.274
	NIC	0.200	NIC	0.188
	describes	0.086	describes	0.158
	Use	0.022	Use	0.068
Describe the rated throughput capacity of a given network medium?	network medium	0.245	rated	0.230
	given network	0.221	network medium	0.215
	capacity	0.221	given network	0.193
	throughput	0.171	capacity	0.193
	rated	0.129	throughput	0.149
	used	0.012	used	0.020

The terms "*network medium*" becomes the most important term in the second question. In the first one although there is no re ordering of the top terms, the most important terms of the question get a larger difference with less important terms. Both enhancements contribute to the retrieval of more relevant documents. The way weights are going to be used is to multiply the statistical weights with them. This is explained more in detail in chapter 3.5.

3.4.4.3 Need for next phase

Having the weights for each term, it is also clear that a list of relevant terms for each query term can also be identified. In information retrieval this is called query expansion. Although the algorithm seems to be able to cope well with the removal of unnecessary terms and weighting of the more important terms, the next investigation will be on including more terms in list of words passed to the Document Retrieval module that are not present in the original query but are related to its terms

3.4.5 Phase 4 - Query expansion

3.4.5.1 Implementation

Apart from using the bigram identification feature developed in the algorithm, the next feature to investigate is the possibility of expanding the query terms from the question. To check how well the query expansion works, the next module corresponding to Document Retrieval is used to evaluate the enhancement brought by Query Expansion. Again using the question *"Why was the OSI model created?"*, the first step is to create a list of the documents that contain both terms "*OSI model*" and "*created*".

The query expansion algorithm developed is based on Vector models, representing each document in vector space as a vector of statistical weights. The method can be described as a reverse classification task. From the current implementation, a list of documents that are related to a query can be retrieved by a strict document selection process where only documents that contain all query terms will be added to the list. From this list, the most frequent terms are selected once the stop words are removed. The weight of each term is then assigned and each document can be represented as a vector of the most frequent terms weights. Since the documents can be treated as a “category” of the query keywords it is safe to assume that the higher weighted terms would be the ones that define the document in vector space.

The initial step is to get frequencies of terms in documents that contain all the query terms. An example of the terms retrieved with their frequencies is shown in Table 3.4-16. The header represents the document id as stored in the database.

Table 3.4-16 – Term frequencies

Term	**Document IDs**			
	854	**857**	**970**	**971**
IP	1	34	22	11
model	12	27	7	8
TCP	1	34	7	11
network(s)	16	13	24	4
OSI	9	15	3	1
layer(s)	1	44	18	12

The next step is for each of the terms to calculate the IDF for each of the terms identified in the previous step. A part of the calculations is shown in Table 3.4-17.

Table 3.4-17 – Term IDF

Term	IDF
IP	2.521784
model	2.788445
TCP	2.779105
network	2.223316
OSI	2.779105
layers	2.838419

For each of the document term, the log frequency is calculated using Equation 3.4-7 for frequencies greater to 0.

Equation 3.4-7 – Log frequency formula

$$LOG\_FREQUENCY = \begin{cases} tf = 0 \rightarrow & 0 \\ tf > 0 \rightarrow & 1 + \log(tf) \end{cases}$$

The results of the calculations are shown in Table 3.4-18

Table 3.4-18 – Log frequency

Term	854	857	970	971
IP	1	2.531479	2.342423	2.041393
model	2.079181	2.431364	1.845098	1.90309
TCP	1	2.531479	1.845098	2.041393
network	2.20412	2.113943	2.380211	1.60206
OSI	1.954243	2.176091	1.477121	1
layers	1	2.643453	2.255273	2.079181

The next step was to apply length normalisation to the weights of each term in a document, using the Equation 3.4-8

Equation 3.4-8 – Length normalisation Log Frequency formula

$$L_2 \frac{= LOG\_FREQUENCY}{\sqrt{\sum LOG\_FREQUENCY_i^2}}$$

Where the Length Normalisation L_2 is Log frequency of the term in the document over the square route of the sum of the squares of the log frequencies of all terms. Part of the results are shown in Table 3.4-19

Table 3.4-19 - Length normalisation of term weights

	Length Normalisation of weighs in documents			
Term	**854**	**857**	**970**	**971**
IP	0.492197	0.738587	0.742774	0.995892
model	1.131582	0.78439	0.646941	1.026595
TCP	0.542421	0.813952	0.644774	1.097512
network	0.956462	0.543769	0.665427	0.689061
OSI	1.060022	0.699684	0.516184	0.537629
layers	0.553998	0.868096	0.804932	1.141686

The final step of the algorithm is to calculate the centroid for each term across the documents that contained the query terms. The results are shown the next section, in table 3.3.5-5

3.4.5.2 Results

In this section, the results for the query expansion enhancement are shown with an analysis of the findings. For a single question the documents that contain the query terms identified by the algorithm so far are the documents with id's 854, 857, 970 and 971. In Table 3.4-20 there is a list of all the higher frequency terms that would be good candidates for query expansion with stop words removed. According to Sahami (2006), important terms will

have centroid over 0.5 where not important terms will have centroid under 0.3. For the specific task, the centroid over 0.5 is used in order to identify the most important terms that define the document collection.

Table 3.4-20 – Centroid weight of term in documents

Term/Document Id	854	857	970	971	Centroid
model	1.131582	0.78439	0.646941	1.026595	0.897377
layers	0.553998	0.868096	0.804932	1.141686	0.8421779
TCP	0.542421	0.813952	0.644774	1.097512	0.774665
IP	0.492197	0.738587	0.742774	0.995892	0.7423627
network	0.956462	0.543769	0.665427	0.689061	0.7136798
OSI	1.060022	0.699684	0.516184	0.537629	0.7033797
reference	0.914623	0.588022	0.518998	0.613723	0.6588414
Internet	0	0.604374	0.620169	0.763838	0.4970953
application	0	0.666099	0.547192	0.741491	0.4886955
transport	0	0.634881	0.552398	0.575348	0.4406568
access	0	0.506976	0.551	0.529136	0.3967778
networking	0.879363	0.545556	0	0	0.3562297
Use	0.465113	0.407252	0.532818	0	0.3512959
used	0.433943	0.457393	0.474975	0	0.3415777
protocol	0	0.683857	0.579963	0	0.3159551
Activity	0.747818	0.509863	0	0	0.3144203
address	0.505098	0	0.748772	0	0.3134676
packets	0	0.620589	0.475593	0	0.2740457
Protocols	0	0.596874	0.340869	0	0.2344358
addresses	0	0	0.727053	0	0.1817631
host	0	0	0.724185	0	0.1810463

From the results, apart from the query terms, the terms *layers* and *reference* appear above the threshold. These terms are related with the

OSI model. On the other hand we have terms such as *TCP/IP* and *network* appearing on the domain definition list.

Adding the term *layer(s)* to the query term list and passing the list through the Document Retrieval module produces the results shown in Table 3.4-21. From the results, it is shown that by controlling the amount of keywords added to the query terms, the result can be improved. One of the main targets of this research is to develop a solution that does not need human intervention to perform the Question Answering tasks. Using the thresholds proposed by Sahami et al (2006), the amount of expanding terms will create noise on the results, since irrelevant terms will be added and the weight of the document will depend on terms not related to the answer. As it is demonstrated in Table 3.4-21, the more terms used to expand the query the greater the weight the Document Retrieval module returned for the wrong document. The document that contains the correct answer is document 854 which is the one with the highest weights using either the sum of the weights of the query terms or the average. Table 3.4-21 also shows the results obtained when the CCNA corpus was the only corpus used (Avg. CCNA and Sum CCNA) as well as when both corpuses described in section 1.6 were used (Avg. REF, Sum REF). Columns three, four and five show the weights using different keywords, which are shown in the header together with their importance weights, which are assigned in as described in section 3.4.4.

Table 3.4-21 – Document weight

		Keywords used in query		
Doc ID	**Weight**	**OSI model(0.590) created (0.409)**	**OSI model(0.46) layers (0.215) created (0.321)**	**OSI model(0.214), IP network(0.208), TCP/IP (0.202), created (0.148), reference (0.129) , layers (0.099)**
854	Avg CCNA	10.387	5.468	1.563
854	Avg REF	2.240	1.788	0.959
854	Sum CCNA	20.774	16.404	9.378
854	Sum REF	4.480	5.363	5.756
857	Avg CCNA	5.836	6.840	4.589
857	Avg REF	1.496	14.047	3.814
857	Sum CCNA	11.672	20.520	27.531
857	Sum REF	2.992	42.142	22.887

3.4.5.3 Need for next phase

The results above show that there is potential for query expansion based on document categorisation techniques, but not in the way they were used in this phase. The problem is that the noise generated can skew the weights and produce unwanted results. This is because adding more terms to the query creates a generalisation of the query term. The questions used are very specific so there is no immediate need to expand the terms. A need of a portable knowledge base based on the different domains the CCNA corpus contains, and the document categorisation techniques like the one described in this section may enhance the Question Answering tasks. In the next section, similar techniques are used in order to categorise terms in two levels, topics and signatures where topics are major domain terms and signature terms are highly related terms to the topic as described in section 2.3.3.

3.4.6 Phase 5 - Topic Signatures

3.4.6.1 Aim

This phase will help answering the research question Q3 "*Is there a categorisation technology such as topic signatures that can be improved from the current state and also used within our algorithm in order to support students*" and will also support hypothesis H4 ("*Topic signatures can be acquired and used for computational tasks, using local analysis techniques and statistical weights without the intervention of an expert user*.").

3.4.6.2 Implementation

Topic signatures have been an inspiration for this research and the potential of this technology to be used in Question Answering systems comes from the ability to create a knowledge base of the corpus with minimum human involvement. In this part of the development, human intervention is replaced by statistical methods. Topic signatures combine local analysis techniques with document categorisation/clustering in order to extract topic and their related signature terms. The steps of the acquisition are:

1. Identify the important terms for each question;
2. From the terms of the previous step, extract topic terms using a statistical metric;
3. Use statistical weights in order to populate the signature terms of each topic;
4. Evaluate the signatures.

For step 1, Inverse Document Frequency (IDF - see Chapter 2.3.2) is used for each of the questions used to evaluate the system. A part of the results is available in the next section 3.4.6.3, with a full evaluation available in

section 4.2.1.4. The threshold of IDF greater than 2 is used to separate a query terms from being a topic term. The selection of this threshold is explained in section 4.2.1.4. If the full corpus is used in order for a topic to be extracted, there is a possibility of non-topic terms to be added to the knowledge base. Although this will drop the precision of the algorithm, it will not affect the performance of the question answering task, because erroneous topics will not be used in questions. For example a highly scoring bigram such as “page” and “concludes” may be picked up as a topic signature, but it’s highly unlikely to be used as an input question to our system. Even if it is entered, there will be no effect on the answer returned. Also erroneous topic signatures that contain stop words will be ignored by the acquisition algorithm.

From the above step, a list of topic terms is generated. For each of the topic a sub-corpus is generated by querying the corpus and retrieving the documents that contain the term. This sub-corpus, or what can be defined as the “elite set” contains all the documents in the corpus that include the topic term. Within this elite set, the log likelihood of each of the terms apart from the topic is calculated. The terms with the higher weight would appear closer to the topic term, and these are considered candidate signature terms. To calculate the log likelihood the following table is used for a term *t* in the elite document set *d*.

	Frequency in d	Frequency in D
Frequency of t	$f_{t,d}$	$f_{t,D}$
Frequency of terms other than t	$f_{t',d}$	$f_{t',D}$

Figure 3.4-2 – Log likelihood observed frequencies

Where:

$f_{t,d}$: Is the frequency of the term occurring in the elite set

$f_{t,D}$: Is the frequency of the term occurring in the learning object not including the documents in the relevant set.

$f_{t,'d}$: Is the frequency of other terms except the one we calculate the weight for in the elite set.

$f_{t',D}$: Is the frequency of other terms except the one we calculate the weight in the learning object not including the documents in the relevant set.

In the following, we will exemplify the technique used to extract signature terms for the topic term “OSI”. The first step would be to create the elite set of documents by retrieving the documents that contain the term *“OSI”*. All the terms other than *“OSI”* are treated as potential signature terms and their frequencies are recorded. For example the frequencies of the words in the elite set are shown in Table 3.4-23 in the results section (3.4.6.3). The same calculation occurs for the non-elite set, which are documents that do not include the term “OSI”. In the non-elite set, only the frequencies of terms that are in the elite set are recorded. Having the individual frequencies of all the non-stop word terms in the elite set, we can use their sum to populate the $f_{t'd}$ value in the table above and using the sum of all non-stop word terms

in the non-elite set, we can derive the $f_{t',D}$ value. A sample of the results is shown in section 3.4.6.3 in Table 3.4-23 and Table 3.4-24. As explained in the results section, there are some terms not related with the term “OSI” near the top of the weight list. To correct this, an extra step is added where a negative log likelihood score is given to a term if there is underuse of a term in the document.

Once the weights are corrected with the overuse/underuse feature, in order for the results to be comparable across documents, normalisation of the weights is required. The reason for normalisation, is that the Log Likelihood ratio provides high scores to words with high probability and lower scores for words with low probability. Because of the use of the “elite-set” we do not have consistency on the reference corpus that we use to compare the terms. So a score of 385 for example as it stands on the *“OSI”* topic and the signature term “*model*” does not mean that the term *“model”* is more significant that a term scoring less for another topic term. Ideally we would want to have the weights ranging from 0 to 1 for the Log Likelihood ratio weights and their sum to be equal to 1 (Moore, R., 2004). To accomplish the above, it is reasonably simple and all is required is to sum all scores and divide each score with the sum. This approach will decrease the confidence of each term with a lower score. An approach to overcome that, is to use the weight of the highest term and divide the rest of the weights by this term.

The results of weight normalisation are shown in Table 3.4-25 of the next section. The topic signature for the topic *“OSI”* is shown in Figure 3.4-3

OSI = {layer:1,model:0.89}

Figure 3.4-3 - "OSI" topic signature

The next section contains the results of this phase, with a minor evaluation.

3.4.6.3 Results

The results of this phase are shown in the tables below.

In Table 3.4-22 given the question "Why was the OSI model created?", the IDF score of each term is identified without using bigrams and removing the stop words. For this calculation the Oxford corpus described in section 1.6.1 was used. The difference in the score for *"OSI"* and the other terms is very large and the reason is that the term *"OSI"* is very specific in the network domain. Using a general corpus to get the usage frequencies makes the topic identification for a learning object that will be mainly dealing with one domain much easier.

Table 3.4-22 – IDF Scores for "Why was the OSI model created?" terms

Keyword	IDF Score
OSI	**Infinity**
model	1.227
created	1.235

As described above, the next step of the algorithm was to identify using the CCNA domain a list of documents that contain the term *"OSI"*. From these documents create a list of terms and their frequencies. The frequency map of terms in the documents that contain the term *"OSI"* is shown in Table 3.4-23

Table 3.4-23 – Term frequencies in elite set

Word	Frequency	Set
network	271	elite
layer	261	elite
data	212	elite
IP	199	elite
Ethernet	148	elite
model	139	elite
address	131	elite
TCP	123	elite
OSI	119	elite

Again using the CCNA corpus and the subset of documents that do not contain the term *"OSI"*, the frequencies of the terms that were picked on the table 3.3.6-2 were measured and shown in Table 3.4-24

Table 3.4-24 - Term frequencies in non-elite set

Word	Frequency	Set
network	606	non-elite
cable	434	non-elite
page	396	non-elite
data	366	non-elite
used	341	non-elite
Ethernet	319	non-elite
address	307	non-elite
Fiber	276	non-elite
IP	263	non-elite

At this stage we have the frequency of the word in the elite set and the frequencies in the non-elite set of potential signature terms. In Table 3.4-25, the Log Likelihood weight of each of the potential signature terms is shown.

Table 3.4-25 – Log likelihood weight for potential signature terms

Term	Log Likelihood
layer	429.3976637
model	385.322603
OSI	324.3452491
cable	166.4642684
TCP	103.8296569
Fiber	87.99572924
IP	66.17600468
application	52.42073582
noise	51.2980107
transport	49.89318238

From the results it is visible that terms like "*cable*" and *"fiber"* have a fairly high score. These terms are not related to the term "OSI". The explanation of that is very simple. The log likelihood measures the surprise element of a condition happening. There are two types of surprise, due to overuse and underuse. To separate these cases, the ratio of the frequency of the term needs to be calculated for each document set (elite/non-elite sets).The surprise used so far was regarding overuse. To separate these two different measures, we multiply with -1 the value of Log Likelihood weight if the usage ratio of the elite set is smaller than the usage ratio of the non-elite set. The ratio is calculated by dividing the frequency of the term over the sum of all terms in the elite and non-elite set.

If the elite set ratio is smaller than the non-elite set ratio, it means that there is a surprise element in our test data, but this surprise is driven by under usage of the term. Applying over usage/under usage logic in the weights of Table 3.4-26 the new weights for the terms are shown in Table 3.4-27.

Table 3.4-26 – Term weight with over/underuse

Term	Log Likelihood with over/under use
layer	429.3976637
model	385.322603
OSI	324.3452491
cable	-166.4642684
TCP	103.8296569
Fiber	-87.99572924
IP	66.17600468
application	52.42073582
noise	-51.2980107
transport	49.89318238

Normalising the weights was the next step of the algorithm described in 3.3.6.1. Two different normalisation methods were used. The first one *LL / Σ (LL)* takes the Log Likelihood weight of a term and divides it by the sum of all log likelihood weights of the potential signature terms. The second normalisation technique *LL / max (LL)* identifies the maximum log likelihood score and then divides all the weights of each potential signature term by the maximum log likelihood weight. From the results in Table 3.4-27, both normalisation techniques seem to provide similar results. The green cells represent terms that are highly linked with the topic terms, while in the red cells contain terms that are not highly correlated with the topic. The blue row is the topic term.

In this section, the results for the task of topic signature acquisition are discussed, with a further evaluation to follow in section 4.2.1.4. The next section contains a summary of the findings from this phase and also a general summary of the module.

Table 3.4-27 – Term weights with normalisation

Term	Log Likelihood with over/under use	LL / Σ (LL)	LL / max(LL)
layer	429.3976637	0.058902696	1
model	385.322603	0.052856693	0.897356077
OSI	324.3452491	0.044492114	0.755349357
TCP	103.8296569	0.014242851	0.241803032
IP	66.17600468	0.009077705	0.154113565
application	52.42073582	0.007190823	0.122079695
transport	49.89318238	0.006844106	0.116193418
Mac	46.80849346	0.006420963	0.10900966
structured	40.68155428	0.005580499	0.094740977
data	34.11395342	0.004679587	0.079446062

3.4.6.4 Findings summary

Topic signatures provide a dynamic knowledge base for our CCNA corpus. Alongside with Bigrams, they represent a way to identify important terms in the domain of any learning object uploaded into a Virtual Learning Environment similar with the CCNA notes used in the experiments. Having this knowledge base, the modules of the next phases are enriched with valuable information inspired from research on document clustering.

The Query Parsing module, as demonstrated in section 4.2.1 can provide quality results that can help any implementation of Document Retrieval and Answer Pinpointing.

The Query Parsing module described in section 3.4, can

- Identify the main terms of the query;
- Check the presence of Bigrams in the query;
- Provide an importance weighting scheme for each term;
- Assign a topic area in a query in order to ensure the answer returned by the Question Answering system is from the same conceptual domain as the query.

The above outcomes are implemented using statistical methods. There are no further enhancements in the Query Parsing module and the outcomes of the module will be passed to the next modules as described in the sections that follow.

3.5 Document Retrieval

The next module of the Question Answering system that is described in this section is the Document Retrieval module. The main responsibility of the module is to use only statistical methods in order to retrieve the answer bearing document from a collection of documents that in our case is the CCNA corpus described in section 1.6. The terms identified by the Query Parsing module (chapter 3.4) are used and only one document that scores highest on a combination of statistical measures will be selected as the one that contains the answer to the query.

The evaluation of this module is simple, since for the pre-defined questions of the CCNA online materials described in section 1.6 the document that contains the answer is pre-defined. After each phase of the Document Retrieval algorithm, the document selected by the algorithm will be compared with the document that has been identified as the one that contains the correct answer. The measure that contains the most correct documents will be the one selected to be used by this module and the performance of the metric using various corpus sizes will also be tested.

A document that contains most of the potential terms has more chance in containing the correct sentences that need to be extracted as well.

We will have two phases included at the development of this algorithm. Phase one will evaluate the term weight parameters, stemming and weighting, whereas in phase two the corpus will be expanded by using the Oxford corpus and the aim is to match the results from Phase 1 in Phase 2. The next section describes Phase 1 of the Document Retrieval development.

3.5.1 Phase 1 – Document retrieval using CCNA corpus

3.5.1.1 Aim

In this phase, we describe the development of an algorithm to extract from a collection of documents the correct document based on the input consisting of the keywords selected in the first module. In the different phases, the type of statistical metric used to measure document weight and the performance in different corpuses is measured. This phase is aimed at answering research question 1 (section 1.4) "*Can a QA system using statistical based techniques provide the similar level of answers as a baseline search engine*" and also support hypotheses H2 *("The correct answer to a question entered to the system should be retrieved using statistical methods and without requiring any background knowledge")*, and H3a ("*A good combination of methods that will work on a learning domain to answer user specific questions is Log Likelihood")*.

3.5.1.2 Implementation

There are five steps in our algorithm, each step using different parameters. The main algorithm begins at step 2 of the list below. Steps 2 and 3 can be described as the potential document selection process. An assumption behind this selection process is that a document containing the most key

terms is more likely to contain the correct sentences that needs to be extracted. Also with the document weighing models used (average and sum features), there is a likelihood for a potential term of medium importance to be repeated often in a document and give a skew the weights.

The main algorithm works as follows:

1. Use terms from Query Parsing module to retrieve documents that contain the terms of the query.
2. From the list of documents, calculate how many instances of the query terms exist in each document.
3. Select the document(s) that contain the maximum number of query terms.
4. For each of the document calculate the document weight.
5. Return the highest scoring document as the document that contains the answer.

In this algorithm there are some parameters that will be set at the different phases such as the statistical measure (TF.IDF or Log Likelihood), the way the document weight is calculated (sum or average of the individual weights). Also the usage of stemming, the Query Parsing module weights and the Topic signatures will be parametrised into the Document Retrieval module

An example of how the potential document selection process will work is described below. Suppose term T_1, which is an important term is included in the query. We have a document D1 that contains T_1 T_2 and T_3 one time each, with weights 10, 6 and 1, respectively. If another document D_2, contains three occurrences of T_2 the Average of the weights for D_1 would be 5.6 while the average weight for D_2 would be 6. The sum of the weights

would be 17 for document D_1, and 18 for D_2. The content of D_2 may not even be relevant to the query, but only be an important document for a part of the query. For this reason a strict selection process is implemented.

The selection process can be described with the example below:

1. Suppose four terms T_1, T_2, T_3 and T_4 are picked by the Query Parsing module.
2. From searching through the corpus, documents D_1 D_2 D3 and D_4 have one or more of the terms from step 1 with the frequencies as shown in Table 3.5-1

Table 3.5-1 – Example selection process

	T_1	**T_2**	**T_3**	**T_4**	**Unique terms**
D_1	1	2	0	1	3
D_2	1	1	0	1	3
D_3	4	1	0	0	2
D_4	0	0	1	0	1

In the case above, the maximum terms that a document should have is 3. Using the documents that contain all the keywords (or maximum number of keywords), only documents D_1 and D_2 will be selected as potential answer bearing documents where their statistical weight would be calculated.

We also run this process using documents that contained not only the maximum number of keyword terms but also documents that contained maximum keywords decreased by one. Using the mock data in Table 3.5-1, documents D_1, D_2 and D_3 are selected for their statistical weighs to be calculated. The problem faced in this case is that in the example above the T_1 term weight multiplied by the statistical weight will make the sum or average of the document higher than the other documents which may be

more relevant. A partial view of the results of this phase is shown in Table 3.5-2

The next objective of this phase is to select a single document from the list of the documents that contain at least one instance of all keyword terms of the query terms. In order for this to be accomplished a full statistical evaluation of the terms in the document list retrieved so far is required.

Table 3.5-2 – Q1 and Q2 document selection

Question: **Why was the OSI model created?**	
Query Module terms: **OSI model, created**	
Document Id	**Unique Terms**
859	2
978	2
967	2
854	2
856	2
938	2
970	2
857	2
971	2
965	2
1016	2

Question: **Why are the pairs of wires twisted together in an UTP cable?**	
Query Module terms: **twisted, UTP cable, pairs, wires**	
Document Id	**Unique Terms**
899	4
888	4
901	4
869	4
870	4
903	4

Each document is then assigned a weight depending on the frequency of the terms within the document. The same term will have different weights in different documents. In the initial stage of this run, for each of the Query Parsing terms the TF.IDF and Log Likelihood values were calculated. The weight of the document is presented as the sum and the average of

individual weights of the Query Parsing terms. The different weighting schemes are adapted by research in different domains of Information Retrieval, so the aim is to identify the best to be used to answer Questions in a Virtual Learning Environment domain. The assumption behind is that if the query terms are present in the document, then the document should contain some information related to our question.

To demonstrate the algorithm, the question *"Why was the OSI model created"* is used; the highest document weights using the sum and average weights (Table 3.4-21) come from the documents 854 and 857 (with the first containing the correct answer). The document 854 is titled "Networking Modes – OSI model" and document 857 is titled "Networking models – TCP/IP". For simplicity, document 854 will be named as OSI-MODEL and 857 as TCP-IP in the tables and calculations that follow.

The OSI-MODEL document contains 371 words grouped into 20 sentences, while the TCP-IP document contains 1000 words formulating 59 sentences. The total number of documents in CCNA corpus is 246.

In Table 3.5-3, the frequencies of each of the query terms in the documents are shown. Just to clarify for this calculation, the bigram identification is not used.

Table 3.5-3 - Term frequencies in documents

DOCUMENT	TERM	FREQUENCY
OSI-MODEL	Created	2
OSI-MODEL	Model	12
OSI-MODEL	OSI	9
TCP-IP	Created	2
TCP-IP	Model	27
TCP-IP	OSI	15

Using "*OSI model*" as a bigram the frequencies are changed as follow (Table 3.5-4).

Table 3.5-4 - Term frequencies in documents using bigrams

DOCUMENT	TERM	FREQUENCY
OSI-MODEL	Created	2
OSI-MODEL	OSI model	4
TCP-IP	Created	2
TCP-IP	OSI model	7

From the tables above, it is evident that the weight of the term "*model*" will place the incorrect document higher in the ranking. Using bigrams, makes the query term more specific and in conjunction with the smaller length of the OSI-MODEL document will raise the document higher in the rank. Gathering more stats, the term "*OSI model*" is present in 54 documents and "*created*" is present in 41 documents in the CCNA corpus.

Using the above stats, the TF-IDF value for the Query Terms can be calculated as below:

$$\text{TF - IDF}_{t,d} = (1 + \log(\text{tf}_{t,d})) \times \log_{10}(N / \text{df}_t)$$

where t is the term and d is the document

TF-IDF $_{\text{(OSI Model, OSI-MODEL)}}$ = (1.602)*0.778 = 1.24

TF-IDF $_{\text{(Created,OSI-MODEL)}}$ = (1.301)*0.658 = 0.85

Sum $_{\text{TF-IDF}}$ = 2.09

Average $_{\text{TF-IDF}}$ = 1.045

TF-IDF $_{\text{(OSI Model, TCP-IP)}}$ = 1.845*0.778 = 1.45

TF-IDF $_{\text{(Created, TCP-IP)}}$ = (1.301)*0.658 = 0.85

Sum $_{\text{TF-IDF}}$ = 2.30

Average $_{\text{TF-IDF}}$ = 1.15

So the document about TCP/IP would weigh higher than the one about the OSI model which is not what we expect. The reason why this happens, is because the documents in CCNA contain information about networking and the main terms are reused throughout the documents. This situation makes it more difficult for domain terms to be able to produce high scores through TF.IDF. Using documents with multiple domains, will make terms like "*OSI model*" stand out more when used because they will only appear in a small subset. Also noteworthy is that the OSI-MODEL document has 4 instances of the bigram *"OSI model"* in the 371 words, while *"OSI model"* appears 7 times in the TCP-IP document which is a relatively bigger document (1000 words). An assumption can be made that OSI-MODEL contains more information about the question than TCP-MODEL. To test that, a statistical

measure that takes into consideration the length of the document and the frequency of the term in the document and the domain (log likelihood) is used. The following calculations explain how log likelihood can be used to assign weights to documents.

The first weight to be calculated is the log likelihood associated to the bigram “OSI model” in the document OSI-MODEL. Table 3.5-5 lists the frequencies of the bigram in the document, in the rest of the corpus and also the frequencies of all other terms.

Table 3.5-5 – Bigram “OSI model” observed frequencies

	Frequencies		
Terms	**OSI-MODEL**	**CCNA**	**Total**
OSI model	4	50	54
Other	367	89448	89815
Total	371	89498	89869

Using the data above, the calculation of log likelihood (Equation 2.3-2) can be achieved by calculating the estimated frequency of the document (E1 - Equation 2.3-3) and the estimated frequency of the domain (E2 - Equation 2.3-4) shown in Table 3.5-6

Table 3.5-6 – Bigram “OSI model” expected frequencies

E1	1.09
E2	0.06
Log Likelihood	15.82

For the term *"created"* on the OSI-MODEL document the following reference table can be generated

Table 3.5-7 – Term "created" observed frequencies

	Frequencies		
Terms	**OSI-MODEL**	**CCNA**	**Total**
created	2	41	43
other	369	89457	89826
Total	371	89498	89869

From Table 3.5-7 the estimated frequencies are calculated which are setting the weight of the term in the document as shown in Table 3.5-8.

Table 3.5-8 - Term "created" expected frequencies

E1	0.54
E2	0.05
Log Likelihood	6.15

So for the document OSI-MODEL the sum of the query term weighs is 21.97 with average weight of terms 10.98

The same calculations are done for the document TCP-MODEL, with the frequencies of the *"OSI model"* and of the other words in the corpus being shown in Table 3.5-9.

Table 3.5-9 - Bigram "OSI model" observed frequencies

	Frequencies		
Terms	**TCP-IP**	**CCNA**	**Total**
OSI model	7	44	51
Other	993	88825	89818
Total	1000	88869	89869

From Table 3.5-10, the estimated frequencies and log likelihood is calculated and the results are available in Table 3.5-10.

Table 3.5-10 - Bigram "OSI model" expected frequencies

E1	0.70
E2	0.05
Log Likelihood	23.25

For the term *"created"* in document TCP-MODEL, the following frequencies are calculated.

Table 3.5-11 - Term "created" observed frequencies

	Documents		
Terms	**TCP-IP**	**CCNA**	**Total**
created	2	41	43
other	998	88828	89826
Total	1000	88869	89869

With the frequencies in Table 3.5-12, the estimated frequencies and log likelihood for the term *"created"* in document TCP-MODEL are calculated in Table 3.5-12

Table 3.5-12 - Term "created" expected frequencies

E1	0.20
E2	0.05
Log Likelihood	2.74

Having the log likelihood of the terms *created* and *OSI model* in both documents, a comparison of the different document weights is shown in Table 3.5-13.

Table 3.5-13 – Document weights comparison

	OSI-MODEL	**TCP-IP**
Average	14.7	12.99
Sum	29.4	25.99

From the table above the document labelled *OSI-MODEL* has greater weight by using the keywords of the question. One feature that was not considered in the specific document is that in the document that contains the correct answer the term "*OSI model*" is also referred as "*OSI reference model*" which will not count towards the frequency of the bigram "OSI model". To overcome this issue a trigram identification algorithm can be added to the existing system but will not solve the problem entirely. For example it will not be possible to identify any potential n-gram and query terms that are not n-grams and they will not benefit from this enhancement.

One approach investigated that produced good results was to check the log likelihood score of each query term with the top two documents that were retrieved. As mentioned before log likelihood is looking for "surprise" occurrences of a term in a text. Having the top two documents compared to each other ensures that the document selected would use the query

terms more than the other. The overuse/underuse feature as described in section 3.4.6.3 is also very important for these calculations.

In the example used so far, the OSI-MODEL document has 371 words and 4 instances of *"OSI model"* whereas the TCP-MODEL has 1000 words with 7 instances of the bigram. This gives log likelihood of 0.45 for document OSI-MODEL with overuse of the term which can be interpreted that there is a small surprise on the usage of the bigram which will be used by our algorithm in order to select OSI-MODEL instead of TCP-MODEL.

Another major point of the development is for the term weighting algorithm to include a stemming module when assigning weights. The difference is very significant as we can see in the evaluation section with Table 4.2-9 showing that the algorithm performs better than the baseline system (Table 4.2-8) by achieving an 80% score instead of the 70% that the baseline produces.

3.5.1.3 Need for next phase

The need for a next phase of development in the Document Retrieval module, comes from the diversity of content expected in Virtual Learning Environments. The system is expected to be able to cope with different size of corpus and also documents from different domains. For this reason the next phase needs to concentrate mainly on an evaluation of the current solution using a larger corpus. In a learning environment, information about the user such as which courses are studied gives some information about what documents the user will have access to. It also indicates some basic categories of the learning objects.

To simulate this scenario and also to evaluate the system on a reference corpus that will provide the algorithm with term frequencies as they are used in a non-networking related document, the Oxford written English corpus (chapter 1.6) is used.

The next section will explain the work related to using the CCNA corpus and the Oxford corpus which will deliver two main outcomes, the first being the response of the system to a bigger corpus and the second highlighting how the performance of the system is affected when a static corpus is used to compare the statistics obtained by the learning object.

3.5.2 Phase 2 – Document retrieval using the CCNA and the Oxford corpus

3.5.2.1 Aim

The aim of this phase is to explore the research question 2 "*How well will our algorithm work using a smaller or a larger corpus since the amount of documents in the VLE will be different per institution.*". This will also support our hypothesis (H2), "*The statistical approaches used should not be dependent on the size of the corpus and the system should be able to retrieve the correct answer having a small or a large corpus to use for weight calculations*".

3.5.2.2 Implementation

In this phase, the Oxford written English corpus, described in chapter 1.6 is used to test the algorithm. This document retrieval enhancement uses a reference corpus in order to compare the usage of terms within our potential answer document with "normal" term usage. Normal term usage is the frequency with which a term appears in texts that are not domain specific.

To separate between the two corpuses and the statistics that are acquired, the CCNA corpus will be also referred to as the domain corpus (DC) where the Oxford one as the static corpus (SC). The domain corpus when the algorithm is used within a VLE will be updated often with new documents, but the static one will be the same and provide term frequencies as used in a good sample of written documents.

To prepare the system, a similar approach is implemented as described in section 1.7. Physically, the data is stored in different tables of the database implemented and although there are some changes in the algorithm for parametrised usage of the two corpuses, switching from the CCNA to the Oxford corpus is easy.

In this phase, only log likelihood based weighs will be used since from the previous step, it is clear that log likelihood performs better. The question *"Which are the two functions of a router in a network?"* is used in order to demonstrate the algorithm.

Running the experiment with the algorithm in the current state (i.e. with the Query parsing module using term weighs, bigrams and stemming and the statistical calculations for the Document Retrieval using Log Likelihood with the Oxford corpus) returns a surprising result. The expected result was to have the domain terms boosting the document weight using the Static Corpus (SC) since there would be a greater element of surprise in the usage of query terms in the domain documents. For the example question used, the document titled *"Networking Terminology - Networking devices"* was picked as the most relevant one. When using only the Domain Corpus

(DC), a different document titled "*IP Routing Protocols - Routing overview"* is selected. The answer expected from the CISCO self-assessment quiz is the one in the *"IP Routing Protocols - Routing overview"* document although both answers can be considered correct from a networking perspective. This raises the issue that looking into multiple documents will increase the quality of the answer.

One of our main aims is that when using either corpuses the answer provided by the system should be the same. To find more about the differences in the weights, further analysis is required in the documents selected, the most relevant using the two different corpuses to identify the reasons why using a different corpus results in different documents as the top documents.

The query terms picked by the Query Parsing module are *functions, router* and *network.* To simplify the frequency tables the document "*IP Routing Protocols - Routing overview"* will be mentioned as ROUTING_OVERVIEW where the document *"Networking Terminology - Networking devices"* will be mentioned as NETWORK_DEVICES in the following calculations. ROUTING_OVERVIEW is the document with the highest document weight when the Domain Corpus (CCNA only) is used, and NETWORK_DEVICES is the document with the highest document weight when the Static Corpus (CCNA and Oxford corpuses) is used.

In the next pages, statistical metrics corresponding to the terms *functions, routers* and *networks* in the documents under investigation are shown, which help with the calculations of the statistical weights. The Term

Frequency in document, is the number of occurrences of the terms in the document, the Term Frequency in learning object is the number occurrences of the term in the full CCNA corpus. The document and learning object length is the number of words in the document and the learning object excluding the document under investigation. The same metrics are calculated using the Oxford corpus.

For the term *functions* in the documents we have the following metrics:

	Documents	
	ROUTING_OVERVIEW	**NETWORK_DEVICES**
Term Frequency in document	2	1
Term Frequency in learning object	43	44
Document length	244	377
Learning object length	49,555	49,422
Term frequency in Oxford corpus	127	127
Oxford corpus length	990,454	990,454
Log Likelihood Domain corpus	5.33	0.84
Log Likelihood Static corpus	12.66	4.15

For the term *router* in the documents we have the following metrics:

	Documents	
	ROUTING_OVERVIEW	**NETWORK_DEVICES**
Term Frequency in document	8	3
Term Frequency in learning object	146	151
Document length	238	375
Learning object length	49,452	49,315
Term frequency in Oxford corpus	0	0
Oxford corpus length	990,454	990,454
Log Likelihood Domain corpus	23.97	2.05
Log Likelihood Static corpus	133.34	52.73

For the term *network* we have the following values:

	Documents	
	ROUTING_OVERVIEW	**NETWORK_DEVICES**
Term Frequency in document	5	20
Term Frequency in learning object	872	857
Document length	241	358
Learning object length	48,726	48,609
Term frequency in Oxford corpus	65	65
Oxford corpus length	990,454	990,454
Log Likelihood Domain corpus	0.10	18.54
Log Likelihood Static corpus	47.22	224.33

What is observed is that for some terms, especially the ones that are used widely in the corpus, their Log Likelihood is quite high.

This is mainly because the frequency density in the reference corpus (or the documents not containing the max amount of keywords) may not be

very high. This in combination with high density in the corpus, will raise the log likelihood.

One of the features used in order to weight query terms is IDF. For the specific query we have the following IDF scores in the corpus of the learning material.

To calculate the IDF the Equation 3.4-2 is used and the scores are shown below:

Term	**IDF in Domain Corpus**
functions	6.69
router	6.38
network	2.22

The table above shows us that the term "*network*" is mainly more general than "*functions*" and "*router*" which is correct. In our learning object we will have many mentions of "*network"*, where the term *"router"* will be in a smaller sub section of the corpus. Similarly the term "*function"*, although a more widely used term in English language in a domain such as the one of the learning object will only be used in documents that contain specific information.

The next step is to normalise the IDF scores to sum 1 (Equation 3.4-4) in order to make them usable with the log likelihood weights. To do that we calculate the sum off all IDF scores (equals to 15.29) and then divide each score by the sum. So the ratio of each term is shown below:

Term	IDF in Domain Corpus
functions	$6.69/15.29 \approx 0.44$
router	$6.38/15.29 \approx 0.42$
network	$2.22/15.29 \approx 0.16$

For each Log likelihood weight we will multiply it with the ratio, in order to give an advantage to terms that seem to be more specific (higher IDF) than others (lower IDF).

This normalisation technique (IDS scores to sum 1) will allow the comparison of the weights independent of factors such as document or corpus length which can create a bias on the weight.

Table 3.5-14 shows the log likelihood weights for each of the query terms (column 1) , using the Domain Corpus(DC) or the Static Corpus (SC) as shown in column 2.The weights are split into two main categories, the ones using the IDF weight (upper table) and the ones not relaying on the IDF weight (lower table). The weight of each term is calculated for both documents ROUTING_OVERVIEW and NETWORK_DEVICES. The final row shows the sum of all term weighs per document and per corpus used.

Table 3.5-14 demonstrates that the weights using the two different corpora can be aligned when the IDF weight is multiplied by the term weight when calculating the document weight as a sum or average of all weights.

Table 3.5-14 – Document weights with and without IDF weighting

		Without IDF weight	
Term	**Corpus**	**ROUTING_OVERVIEW**	**NETWORK_DEVICES**
functions	DC	5.33	0.84
	SC	12.66	4.15
router	DC	23.97	2.05
	SC	133.34	52.73
network	DC	0.1	18.54
	SC	47.22	224.33
Document Weight	DC	**29.4**	**21.43**
	SC	**193.22**	**281.21**
		With IDF weight	
Term	**Corpus**	**ROUTING_OVERVIEW**	**NETWORK_DEVICES**
functions	DC	2.35	2.35
	SC	12.66	2.35
router	DC	10.07	0.86
	SC	56.00	22.15
network	DC	0.02	2.97
	SC	7.56	35.89
Document Weight	DC	**12.43**	**6.17**
	SC	**76.22**	**60.38**

3.5.2.3 Need for next phase

No further phases are needed for this part of the system. The results from Table 3.5-14 demonstrates that with the usage of IDF term weight in the Domain Corpus, brings out the importance of the term in the domain, and at the same time avoids skewing the document weight. Also the use of a reference static corpus provides the system with word frequencies as used in general written documents which makes surprises easier to identify.

The next module that will be described in the next section is the Answer Pinpointing module. This module receives the document selected by the current module and then returns a subset of sentences as the answer to the user question.

3.6 Answer Pinpointing

3.6.1 Aim

The aim of this part of the development is to use the document that scored highest in the Document Retrieval module and extract part of it in order to compose the answer. The baseline system will return a list of documents since so far only search engines are used in Virtual Learning Environments to support the learner when looking for query based information. This run will help answering the research question Q4 *("Summarisation is an Information Retrieval technique that, given a document, returns the important sentences of a document. Would that kind of technique be of use for choosing the answer to a user question?")*, and support hypothesis H3b ("*A good combination of methods that will work on a learning domain to answer user specific questions involves Summarisation techniques - to extract sentences relevant to the user query*")

3.6.2 Implementation

There were two phases involved in the development of the answer pinpointing module. The first one was very simplistic. It can be interpreted as an enhancement to the existing search systems that return a list of documents sorted by document weight. The initial implementation returns the full top weighted document that is identified from the Document Retrieval module. This can cause a problem if an answer requires text

extracts from multiple documents, but such questions not within the scope of our research, which is to provide better responses than current Virtual Learning systems.

Once providing a better solution was achieved, the next phase in this implementation concentrated on trying to apply statistical techniques using any resources available to limit the amount of sentences passed as an answer to the user. The basic approach supporting our algorithm uses document summarization (Lam-Adesina, Jones, 2001)

To demonstrate the algorithm, the question *"What is required for electrons to flow?"* is used. Asking the Question Answering system at the current state this question, the answer is shown in Table 3.6-1 where each sentence is represented as a row of the table. The first column contains a unique numeric *id* for each sentence, the second column contains the actual sentence text and the third column shows the index of the sentence inside the document. The set of sentences that can be returned as correct answer are highlighted with green. Also in the table any instance of the query terms (*"electron"* and *"flow"*) that are picked by the Query Parsing module are highlighted with blue and yellow.

The main aim is to develop an algorithm that will pick sentences with id's 20184-20195 or 20225-20234 and remove the other sentences from the answer. This will increase the precision of the answer provided from the QA system in comparison with the answer given by baseline systems which would be a list of documents.

Table 3.6-1 – Keyword term frequency per sentence

20183	Copper Media Circuits This page explains circuits.	0
20184	Current flows in closed loops called circuits.	1
20185	These circuits must be made of conductive materials and must have sources of voltage.	2
20186	Voltage causes current to flow.	3
20187	Resistance and impedance oppose it.	4
20188	Current consists of electrons that flow away from negative terminals and toward positive terminals.	5
20189	These facts allow people to control the flow of current.	6
20190	Electricity will naturally flow to the earth if there is a path.	7
20191	Current also flows along the path of least resistance.	8
20192	If a human body provides the path of least resistance, the current will flow through it.	9
20193	When an electric appliance has a plug with three prongs, one of the prongs acts as the ground, or 0 volts.	10
20194	The ground provides a conductive path for the electrons to flow to the earth.	11
20195	The resistance of the body would be greater than the resistance of the ground.	12
20196	Ground typically means the 0-volts level in reference to electrical measurements.	13
20197	Voltage is created by the separation of charges, which means that voltage measurements must be made between two points.	14
20198	A water analogy can help explain the concept of electricity.	15
20199	The higher the water and the greater the pressure, the more the water will flow.	16
20200	The water current also depends on the size of the space it must flow through.	17
20201	Similarly, the higher the voltage and the greater the electrical pressure, the more current will be produced.	18
20202	The electric current then encounters resistance that, like the water tap, reduces the flow.	19
20203	If the electric current is in an AC circuit, then the amount of current will depend on how much impedance is present.	20
20204	If the electric current is in a DC circuit, then the amount of current will depend on how much resistance is present.	21

20205	The pump is like a battery.	22
20206	It provides pressure to keep the flow moving.	23
20207	The relationship among voltage, resistance, and current is voltage (V) equals current (I) multiplied by resistance (R).	24
20208	In other words, V=I*R.	25
20209	This is Ohm's law, named after the scientist who explored these issues.	26
20210	Two ways in which current flows are alternating current (AC) and direct current (DC).	27
20211	AC voltages change their polarity, or direction, over time.	28
20212	AC flows in one direction, then reverses its direction and flows in the other direction, and then repeats the process.	29
20213	AC voltage is positive at one terminal, and negative at the other.	30
20214	Then the AC voltage reverses its polarity, so that the positive terminal becomes negative, and the negative terminal becomes positive.	31
20215	This process repeats itself continuously.	32
20216	DC always flows in the same direction and DC voltages always have the same polarity.	33
20217	One terminal is always positive, and the other is always negative.	34
20218	They do not change or reverse.	35
20219	An oscilloscope is an electronic device used to measure electrical signals relative to time.	36
20220	An oscilloscope graphs the electrical waves, pulses, and patterns.	37
20221	An oscilloscope has an x-axis that represents time, and a y-axis that represents voltage.	38
20222	There are usually two y-axis voltage inputs so that two waves can be observed and measured at the same time.	39
20223	Power lines carry electricity in the form of AC because it can be delivered efficiently over large distances.	40
20224	DC can be found in flashlight batteries, car batteries, and as power for the microchips on the motherboard of a computer, where it only needs to go a short distance.	41
20225	Electrons flow in closed circuits, or complete loops.	42

20226	Figure shows a simple circuit.	43
20227	The chemical processes in the battery cause charges to build up. This provides a voltage, or electrical pressure, that enables electrons to flow through various devices.	44
20228	The lines represent a conductor, which is usually copper wire.	45
20229	Think of a switch as two ends of a single wire that can be opened or broken to prevent the flow of electrons.	46
20230	When the two ends are closed, fixed, or shorted, electrons are allowed to flow.	47
20231	Finally, a light bulb provides resistance to the flow of electrons, which causes the electrons to release energy in the form of light.	48
20232	The circuits in networks use a much more complex version of this simple circuit.	49
20233	For AC and DC electrical systems, the flow of electrons is always from a negatively charged source to a positively charged source.	50
20234	However, for the controlled flow of electrons to occur, a complete circuit is required.	51
20235	Figure shows part of the electrical circuit that brings power to a home or office.	52
20236	The Lab Activity explores the basic properties of series circuits.	53
20237	The next page covers cable specifications.	54
20238	Lab Activity Lab Exercise: Series Circuits In this lab, the student will build and explore the basic properties of series circuits.	55
20239	Web Links	56

Analysing the sentences highlighted with light green, which can be accepted as the correct answer, it is noticeable that the correct answers have a high concentration of the query terms. For example sentences with IDs between 20235-20239 do not contain any of the query terms. Sentences with IDs between 20196 and 20224 do not contain the term “electron”. The sentence clusters that contain the correct answer do have sentences that contain one or more keywords with a maximum of one

sentence that does not have a query term in a window of three sentences (such as sentences 20192-20194).

The algorithm used contains the following steps.

1. Pick the highest ranking document from the Document Retrieval module;
2. Use the keywords that have been extracted by the Query Parsing module as boundaries for the answer;
3. Iterate through the sentences and split the document into answer clusters using the rules below:
    a. A cluster starts each time a sentence has a keyword frequency greater than one.
    b. While the sentences have a keyword frequency greater than 0 we add the sentence to the potential answer cluster.
    c. Suppose the keyword frequency of the current sentence with index *n* is zero, we check the keyword *frequency* of the sentence with index *n+1*.
    d. If the keyword frequency of the sentence with index *n+1* is zero as well, we end the potential answer cluster at index n-1 and continue extracting clusters from the next sentences in the document.
    e. If the keyword frequency of the sentence with the index *n+1* is greater than zero, then we add both sentences to the cluster.
4. For all the potential clusters that are identified in step 3, calculate the sum of all frequencies of the sentences in the cluster and divide them

by the number of sentences in the cluster. This is the weight of the cluster.

5. Return the cluster with the greatest weight as the answer to the question.

Using the algorithm above, the clusters that are created are the following:

Cluster 1			
ID	**Sentence text**	**Index**	**f**
20184	Current flows in closed loops called circuits.	1	1
20185	These circuits must be made of conductive materials and must have sources of voltage.	2	0
20186	Voltage causes current to flow.	3	1
20187	**Resistance and impedance oppose it.**	**4**	**0**
20188	Current consists of electrons that flow away from negative terminals and toward positive terminals.	5	2
20189	These facts allow people to control the flow of current.	6	1
20190	Electricity will naturally flow to the earth if there is a path.	7	1
20191	Current also flows along the path of least resistance.	8	1
20192	If a human body provides the path of least resistance, the current will flow through it.	9	1

Cluster 2			
ID	**Sentence text**	**Index**	**f**
20199	The higher the water and the greater the pressure, the more the water will flow.	16	1
20200	The water current also depends on the size of the space it must flow through.	17	1

Cluster 3			
ID	**Sentence text**	**Index**	**f**
20225	Electrons flow in closed circuits, or complete loops.	42	2
20226	**Figure shows a simple circuit.**	**43**	**0**
20227	The chemical processes in the battery cause charges to build up. This provides a voltage, or electrical pressure, that enables electrons to flow through various devices.	44	2
20228	**The lines represent a conductor, which is usually copper wire.**	**45**	**0**
20229	Think of a switch as two ends of a single wire that can be opened or broken to prevent the flow of electrons.	46	2
20230	When the two ends are closed, fixed, or shorted, electrons are allowed to flow.	47	2
20231	Finally, a light bulb provides resistance to the flow of electrons, which causes the electrons to release energy in the form of light.	48	3
20232	**The circuits in networks use a much more complex version of this simple circuit.**	**49**	**0**
20233	For AC and DC electrical systems, the flow of electrons is always from a negatively charged source to a positively charged source.	50	2
20234	However, for the controlled flow of electrons to occur, a complete circuit is required.	51	2

In the clusters, sentences with no query terms are highlighted in bold font but they are surrounded by sentences that contain query terms.

The weight of a cluster is calculated by summing the frequencies of the query terms and dividing the sum by the length of the cluster. So for Cluster 1 will give a weight of 8/10. Cluster 2 has a weight of 2/2 where Cluster 3 has a weight of 15/10. The length of a cluster would be the amount of

sentences it contains. So Cluster 1 is of length 9, Cluster 2 is of length 2 and Cluster 3 has length of 10. A full evaluation of the Answer Pinpointing module is available in chapter 4.2.3.

3.7 Limitations of methodology

The main weakness of the algorithms described in chapters 3.4, 3.5 and 3.6 is if an answer is spread across more than one document, our algorithms will not be able to pick the correct answer. This is something that is introduced by design since our hypotheses do not investigate multi document answers. Providing an answer assembled from multiple documents is something to be considered for future developments and is discussed in more detail in the conclusions.

Another limitation comes from one of the external tools, is that the sentence splitter may not support multiple languages. This issue can also be rectified by using document locale information and then by using a library of sentence splitters that can be selected depending on the language.

On the topic of multiple languages, the stop word list used by the Query Parsing module is another area that introduces a limitation through a dependency on an external tool. Again this can be fixed by using multiple stop word lists and a language identifier module.

3.8 Database entries and schema

The HTML files described in section 1.6 were parsed into sentences and words and then saved into a database shown in Figure 3.8-1. This step is implemented in order to be able to perform calculations quicker and also

instead of working on the text extracted from the learning object we can use a cached version stored into a database.

The database structure holds all the domain objects we will need to calculate statistics of. Each document is stored in the table *DOCUMENT*. We can create any sub corpus of documents need to derive various statistical weights (such as local feedback), and we can also select documents using specific keywords. The next important entity from a statistical point of view is a sentence. Every sentence in a corpus document is stored in a table *SENTENCE* which is linked to the document table via the *DOCUMENT_SENTENCE* table. This way individual sentences can be picked that contain keywords, get some statistical metrics on a term or a sentence and also retrieve all the sentences for a document. The final entity we need is each term which is going to be stored into two tables. The table WORD that contains the string representation of a term and also the stemmed string for this term that uses Lucene's internal implementation to retrieve the stem of a word. Also the table *SENTENCE_WORD* contains the work_pk1 (unique identified of a word) that links the entity of a sentence word with a sentence.

With this design, occurrences of a word in a document or in a sentence or within a corpus or a sub corpus can be retrieved using easy SQL queries. There is also information on neighbour terms which enables us to quickly identify bi-grams and their weights.

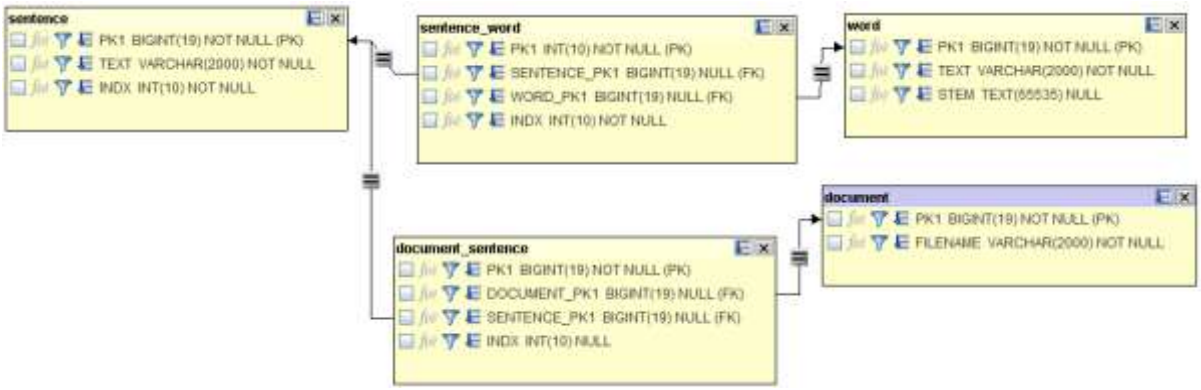


Figure 3.8-1 – Database schema

3.9 Technology breakdown of modules

In this section the final choice of technologies that are used in the modules of system.

One a question reaches the Query parsing module, stop words are removed using a list and also the remaining potential terms are stemmed in order to reduce terms to a common stem.

Afterwards, the query parsing module uses bigrams identified in the pre-processing stage to identify any terms that will be more significant in the next stages. The IDF of all the terms is also calculated and normalised to sum 1, in order to have an importance based attribute of the keyword terms.

The document retrieval module is using a document filtering submodule, where is only uses documents that contain the max amount of query keywords subtracted by one. So if the a subset of retrieved documents contain a max of n keyword terms, then the algorithm will try to look for the answer in the subset of documents that contain n and n-1 keywords.

On the selected subset of documents described in the paragraph above, we calculate the sum an average Log Likelihood of the keyword terms. The document with the max sum and average Log Likelihood is selected as the one that contains the answer.

Finally the Answer pinpointing document, is using the algorithm that is described in section 3.6.2.

A diagram of the final system is shown in Figure 3.9-1Figure 3.9-1

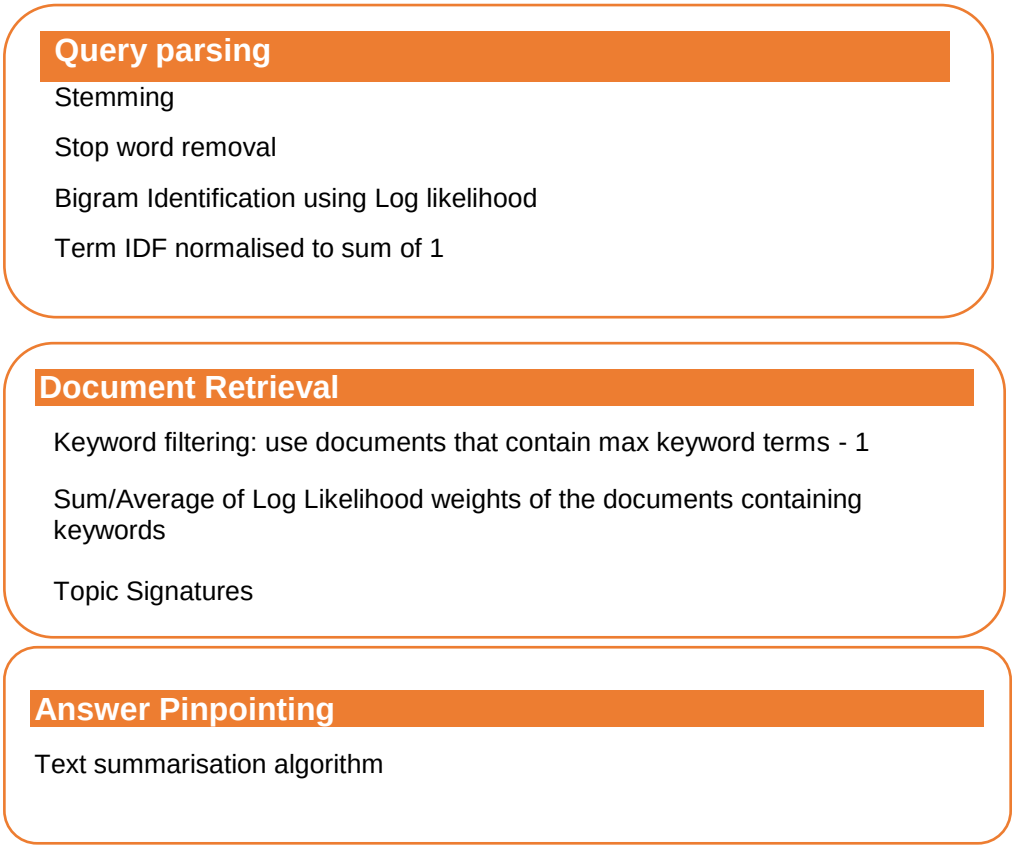


Figure 3.9-1 – Detailed technologies

Also a detailed process list of each module is shown in

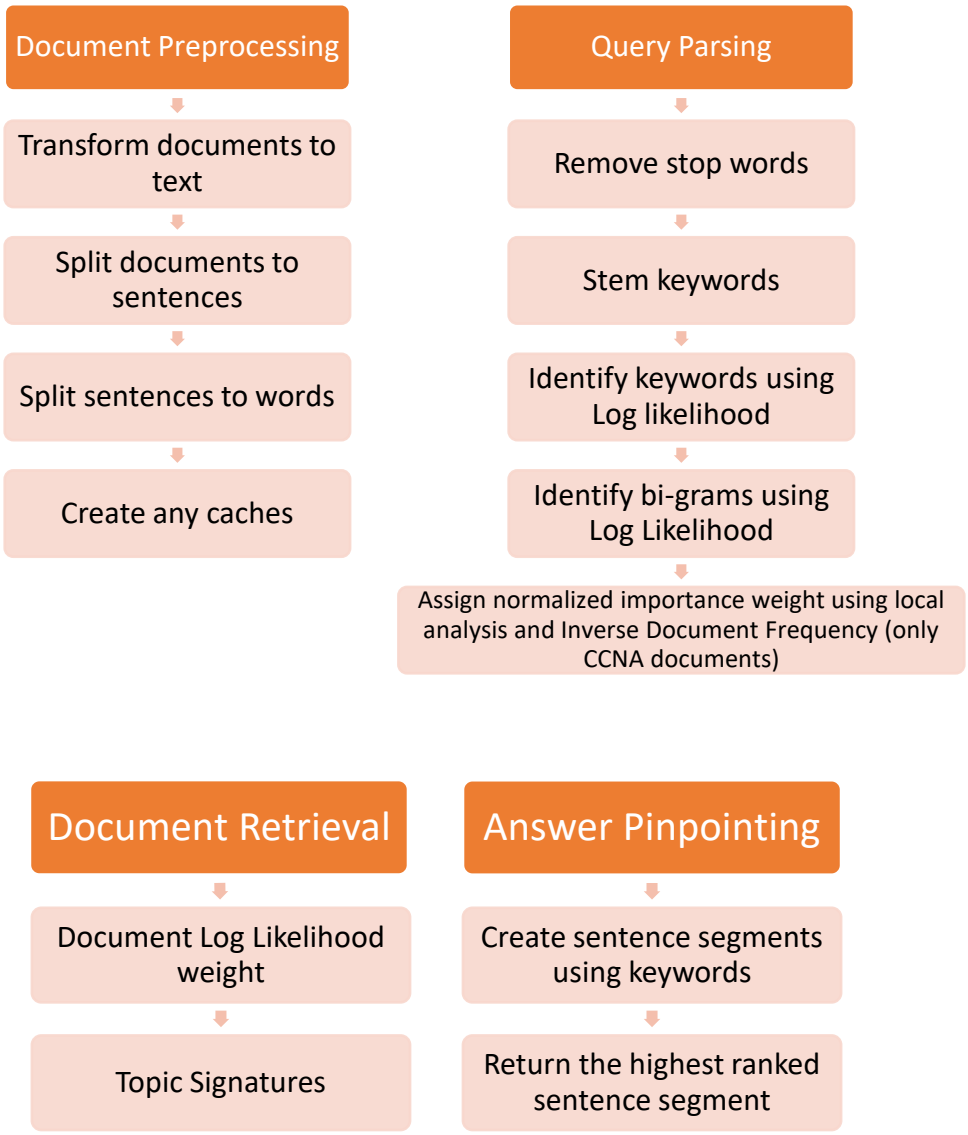


Figure 3.9-3.9-2 - Process breakdown

Chapter 4

4 Evaluation - Research findings

4.1 Introduction

In this chapter we will discuss the research findings from the different phases of the development as per the methodology. We will split this discussion in two main sections. The first section 4.2, focuses on the system evaluation which are experiments that we conducted to measure the performance of the actual system and to compare it with the performance of a baseline system. The next section (4.3) will describe the user evaluation and will present the results of the experiment we conducted with a student sample.

4.2 System Evaluation

The evaluation will be divided according to the type of the experiments that were conducted. There are two main categories of experiments, the ones conducted without users, referred to as System Evaluation, and the experiments that were conducted with users and are referred to as User Evaluation. In this area we will describe the results of our experiments using the data we have acquired from the list of the questions shown in Table 4.2-1.

Table 4.2-1 – Self Assessment Questions

Q1	Describe the use of a network interface card (NIC)?
Q2	Describe the rated throughput capacity of a given network medium?
Q3	What describes a LAN?
Q4	Why was the OSI model created?
Q5	Why are the pairs of wires twisted together in an UTP cable?
Q6	What is required for electrons to flow?
Q7	How does using a hub or a repeater affect the size of the collision domain?

Q8	What can cause a collision on an Ethernet network?
Q9	Which Ethernet implementations use rj-45 connectors?
Q10	Which are the functions of a router in a network?

For question answering we have only used the CCNA online notes and if a reference corpus is required, the Oxford Written English corpus is used.

4.2.1 Query parsing module

The aim of this module is to retrieve the main terms of a question/query and if necessary expand it with relevant terms. The way to evaluate the extraction algorithms either by using the stop word list or the local feedback would be to assess the following metrics

- Non-stop words extracted: This measures the amount of non-stop words that were identified by the Query Parsing module.
- Non-stop words in question: This measures the amount of non-stop words that are available in the question.

4.2.1.1 Phase 1 – Pilot Run

In this run, we only used the stop word list in order to retrieve terms. Each query was split into an array of words and each word individually was checked against a static stop word list.

From the results in Table 3.4-1 it can be seen that there is only a small level of improvement that can be made in relation to removing stop words. We can see that for question 7 we have a small drop in precision. This is because the stop word list does not contain the term *does*. Using the stem of the potential keyword would be sufficient to sort out this issue.

Also in the case of question 9, the term *RJ-45* is not picked up as a term. This error is due to how we parse the keywords before adding them to the database, where any non-alphanumeric characters are removed. This also leads to the next phase of this module that adds bigram information to improve Query Parsing.

4.2.1.2 Phase 2 - Bigram Evaluation

In this phase we wanted to evaluate if there were any improvements on picking up bigrams from the query/question we entered in the system. In this experiment we want to ensure that all bigrams are correctly identified. The results are shown in Table 4.2-2.

Table 4.2-2 – Query parsing phase 2 results

Question	Extracted terms	Number of extracted bigrams	Number of bigrams	Bigram precision	Bigram recall
Describe the use of a network interface card (NIC)?	network interface interface card NIC following use describes	2	2	100%	100%
Describe the rated throughput capacity of a given network medium?	given network medium used capacity throughput rated	1	1	100%	100%
What describes a LAN?	LAN describes	0	0	-	-
Why was the OSI model created?	OSI model created	1	1	100%	100%
Why are the pairs of wires twisted together in an UTP cable	UTP cable wires twisted pairs	1	1	100%	100%
What is required for electrons to flow?	required flow electrons	0	0	-	-
How does using a hub or a repeater affect the size of collision domain?	collision domain does using size repeater hub affects	1	1	100%	100%

Question	Extracted terms	Number of extracted bigrams	Number of bigrams	Bigram precision	Bigram recall
Which of the following will cause a collision on an Ethernet network?	Ethernet network following cause collision	1	1	100%	100%
Which Ethernet implementations use rj-45 connectors?	Ethernet implementations Use connectors	1	2	100%	50%
What are the functions of a router in a network?	network router functions	0	0	-	-

As it is shown, there is a drop in recall for question 9. The reason behind that is that the term RJ-45 has not being identified as a potential term which creates this exclusion. Apart from that, our results are very accurate and the module seems to be working at a satisfactory level since for the majority of questions, the bigram identification is performed with 100% precision.

4.2.1.3 Phase 3 - Term weights.

The next set of results we will be evaluating concerns the term weights assigned by the Query Parsing module. The evaluation of the individual scores is not very important, since one cannot confidently measure the importance of a term down to decimal digits, but with the percentage score we can see the distribution of importance for each term. The results are shown in Table 4.2-3. For example, if there are two keywords in a question both weighting around the 0.5 mark, like in the case of question 3, then we know that both terms are equally important for the query. In questions like Q6 we can see that the main weight of the query is placed on into the *electrons* and the rest of the terms share the other 50% of the "*importance*".

Table 4.2-3 - Query parsing phase 3 results

Question Number	Term	%weight		Question Number	Term	%weight
Q1	interface card	0.311		**Q6**	electrons	0.508
	network interface	0.274			flow	0.275
	NIC	0.188			required	0.218
	describes	0.158		**Q7**	affects	0.205
	Use	0.068			repeater	0.181
Q2	rated	0.230			Hub	0.164
	network medium	0.215			collision domain	0.157
	given network	0.193			size	0.128
	capacity	0.193			using	0.083
	throughput	0.149			does	0.083
	used	0.020		**Q8**	Ethernet network	0.371
Q3	describes	0.587			cause	0.337
	LAN	0.413			collision	0.291
Q4	created	0.511		**Q9**	Ethernet implementations	0.496
	OSI model	0.489			connectors	0.385
Q5	twisted	0.277			use	0.119
	UTP cable	0.245		**Q10**	functions	0.492
	pairs	0.239			router	0.413
	wires	0.239			network	0.095

The results of this table are discussed individually since the precision/recall of the algorithm cannot be easily calculated.

For Q1 70% of the weight is associated the term “Network Interface card” or “NIC”. So we know that this is the main term of the query. Our limitation of the system arises from it being unable to pick trigrams and introduces an issue with the double bigram “network interface”/”interface card”. Also the

term “NIC” is an acronym for the trigram. The weights of the other two terms are quite low in the range of 10% and 6%. We can accept the weighting of the query terms as correct for this question.

Q2 seems not to have any standout terms, since 80% of the measured importance is distributed across 4 terms. The term weighting though for this specific question is still correct having 4 important terms followed by the term “*throughput*”. The term “*used*” does not carry specific meaning and rates last. Again this would be an acceptable result for the module.

Moving to Q3 and Q4, the two important terms for each question are almost equal in weight in the region of 50%. Verbs score a bit higher than the noun which will carry most of the meaning.

In Q5 we have four equally important words in the query and this is again true.

In the case of question 6, the term “*electrons”* dominates the importance of the query with a 50% score and the rest of the weight is split between the remaining terms which is an acceptable weight distribution.

In question 7 there is a slight difference from what is expected. The bigram *“collision domain”* and the terms “*hub*” and “*repeater*” are expected to score higher on the ranking. In this case they do not, but the distribution of weights is quite equal. Also some stop words that have not been picked so far score low. This can prove that even if they get a high score from the Document Retrieval module, applying the weight from the Query Parsing module (0.083) will make their overall score smaller and will not affect the result.

In Q9, again the main terms of the query are splitting the importance which is satisfactory.

Finally, in Q10, the two terms, "*functions"* and "*routers"*, have taken more than 90% of the importance of the query with the term "*network*" scoring as low as 9%. The reason behind that is that the term "*network"* appears very often in our learning object so it cannot be treated as a special term.

Using stemming the same algorithm creates the results shown in Table 4.2-4

Table 4.2-4 - Query parsing phase 3 results with stemming

Question Number	Term	%weight		Question Number	Term	%weight
Q1	interface card	0.368		**Q6**	electrons	0.436
	network interface	0.324			flow	0.379
	NIC	0.200			required	0.185
	describes	0.086		**Q7**	collision domain	0.203
	Use	0.022			affects	0.187
Q2	network medium	0.245			Hub	0.172
	given network	0.221			repeater	0.166
	capacity	0.221			size	0.149
	throughput	0.171			does	0.107
	rated	0.129			using	0.016
	used	0.012		**Q8**	Ethernet network	0.410
Q3	LAN	0.564			collision	0.303
	describes	0.436			cause	0.287
Q4	OSI model	0.591		**Q9**	Ethernet implementations	0.570
	created	0.409			connectors	0.392
Q5	twisted	0.315			Use	0.037
	UTP cable	0.279		**Q10**	functions	0.459
	pairs	0.238			router	0.453
	wires	0.168			network	0.088

One of the main improvements is that the verbs in questions 3 and 4 score less than the main terms. Also stop words like *use*, *does* and *describes* score much lower than on the run without the stemming enabled. Stemming created a positive effect on the results and is an integral part of the term weighting module since there is a significant improvement in setting the term weights.

4.2.1.4 Phase 5 - Topic signatures

In this section results for selecting relevant topic terms and the retrieval of signature terms are presented. A simple IDF metric is used to pick up the most important query terms. The results of this experiment are shown below. The first row of the table corresponds to the question number as presented in Table 4.2-1. The row labelled "keyword", contains the keywords extracted following stop word removal and they are shown in Table 3.4-1 of the Query Parsing Evaluation. Bigram identification is not used since the topic term is a single word. The third row captures the IDF value as described in Equation 3.4-2. In the *Topic* column, **Y** is added if, after manual evaluation, the term is identified to be a potential topic. In the fifth column (*Topic and picked*) we set the value to TRUE if the term has an IDF greater than 2. The next column "T*opic and not picked*" is set to TRUE when the keyword is manually picked as a topic but the IDF score is below the IDF baseline. The final column will have a TRUE value if the keyword is picked (IDF>2) but the term is not considered a topic.

The threshold for selecting topic terms is set to IDF greater than 2 (this was initially picked empirically, because it depends on the size of corpus). To get a log of 2, the result obtained by dividing the total number of documents

to the number of documents containing the term should be greater than 100. So given our reference corpus of 853 documents, a word would need to appear in maximum of 8.5 documents.

Table 4.2-5 – Topic extraction

Question	Keyword	IDF Score	Topic	Topic and picked	Topic and not Picked	Not topic and picked
Q1	**network**	**2.622**	Y	TRUE	FALSE	FALSE
	interface	**4.264**	Y	TRUE	FALSE	FALSE
	card	**4.264**	Y	TRUE	FALSE	FALSE
	NIC	**Infinity**	Y	TRUE	FALSE	FALSE
	use	0.463	N	FALSE	FALSE	FALSE
	describes	1.897	N	FALSE	FALSE	FALSE
Q2	given	0.583	N	FALSE	FALSE	FALSE
	network	**2.622**	Y	TRUE	FALSE	FALSE
	medium	**2.500**	Y	TRUE	FALSE	FALSE
	used	0.359	N	FALSE	FALSE	FALSE
	capacity	1.779	Y	FALSE	TRUE	FALSE
	throughput	**5.139**	Y	TRUE	FALSE	FALSE
	rated	3.858	Y	TRUE	FALSE	FALSE
Q3	**LAN**	**Infinity**	Y	TRUE	FALSE	FALSE
	describes	1.897	N	FALSE	FALSE	FALSE
Q4	**OSI**	**Infinity**	Y	TRUE	FALSE	FALSE
	model	1.227	Y	FALSE	TRUE	FALSE
	created	1.235	N	FALSE	FALSE	FALSE
Q5	**UTP**	**Infinity**	Y	TRUE	FALSE	FALSE
	cable	**4.957**	Y	TRUE	FALSE	FALSE
	wires	**4.669**	Y	TRUE	FALSE	FALSE
	twisted	**5.362**	Y	TRUE	FALSE	FALSE
	pairs	**3.571**	Y	TRUE	FALSE	FALSE

Question	Keyword	IDF Score	Topic	Topic and picked	Topic and not Picked	Not topic and picked
Q6	required	0.908	N	FALSE	FALSE	FALSE
	flow	1.995	Y	FALSE	TRUE	FALSE
	electrons	**3.804**	Y	TRUE	FALSE	FALSE
Q7	**collision**	**5.139**	Y	TRUE	FALSE	FALSE
	domain	**2.590**	Y	TRUE	FALSE	FALSE
	does	0.405	N	FALSE	FALSE	FALSE
	using	0.655	N	FALSE	FALSE	FALSE
	size	1.643	Y	FALSE	TRUE	FALSE
	repeater	**Infinity**	Y	TRUE	FALSE	FALSE
	hub	**5.650**	Y	TRUE	FALSE	FALSE
	affects	**2.039**	Y	TRUE	FALSE	FALSE
Q8	**Ethernet**	**6.749**	Y	TRUE	FALSE	FALSE
	network	**2.622**	Y	TRUE	FALSE	FALSE
	following	0.775	N	FALSE	FALSE	FALSE
	cause	0.987	N	FALSE	FALSE	FALSE
	collision	**5.139**	Y	TRUE	FALSE	FALSE
Q9	**Ethernet**	**6.749**	Y	TRUE	FALSE	FALSE
	implementations	**4.803**	Y	TRUE	FALSE	FALSE
	Use	0.463	N	FALSE	FALSE	FALSE
	connectors	**5.362**	Y	TRUE	FALSE	FALSE
Q10	**network**	**2.622**	Y	TRUE	FALSE	FALSE
	router	**Infinity**	Y	TRUE	FALSE	FALSE
	functions	1.874	N	FALSE	FALSE	FALSE

To calculate the precision of the above method, the precision equation (Equation 2.4-1) is used.

$$\text{Precision} = \frac{29}{29} = 1 = 100\%$$

For the recall, the recall formula (Equation 2.4-2) is used.

$$\text{Recall} = \frac{29}{33} = 0.879 = 87.9\%$$

Finally the F-Score is calculated using Equation 2.4-3 – F-score

$$\text{F score} = \frac{2\times(1\times0.879)}{(1+0.879)} = 0.935$$

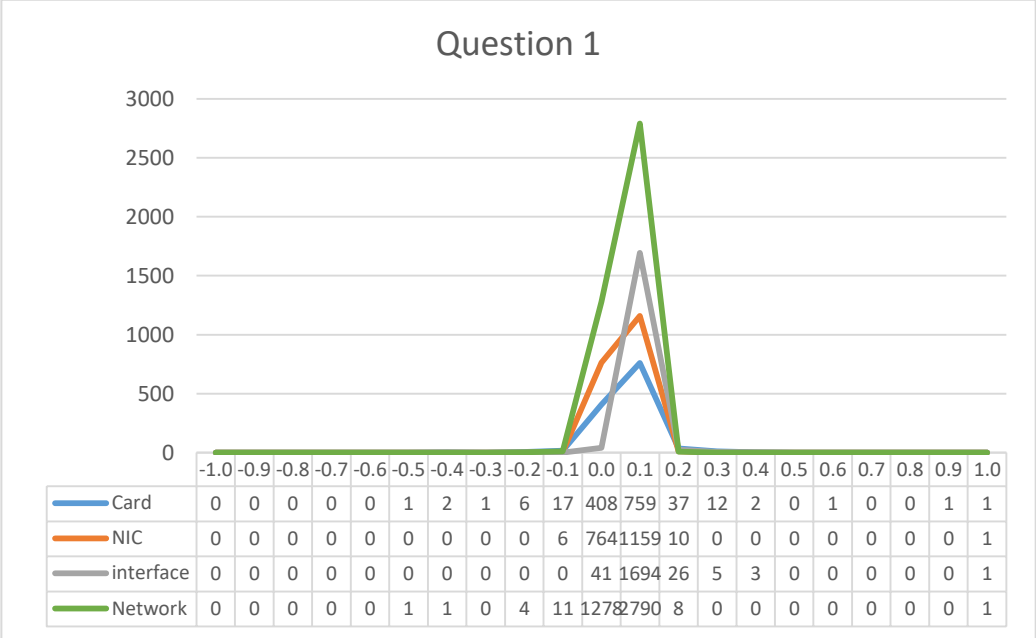
The next step is to display the results of identifying the signature terms. As described in the methodology section (3.4.6), the Log Likelihood score for each of the words in the documents that a topic term appears in is calculated. The score is then normalised by dividing them with the one the topic term has since this would be the highest. Log Likelihood overuse/underuse, where if the usage ratio of the elite set is smaller than the usage ratio of the non-elite set, then the Log Likelihood score is multiplied by -1, is also used and shown on the negative X axis.

In this section, for each topic term we will show the distribution of weights, which is proven to be similar to a normal distribution, and also we will check different σ boundaries in order to identify any signature terms outside normal distribution. This will evaluate how well different boundaries work with the selection algorithm.

For each of the topic terms identified above, we calculate the weight distribution for all the other terms in the document. The terms that fall outside of the 95% (two standard deviations from the mean) of the

distribution are automatically added as signature terms. The results for each signature term are shown below

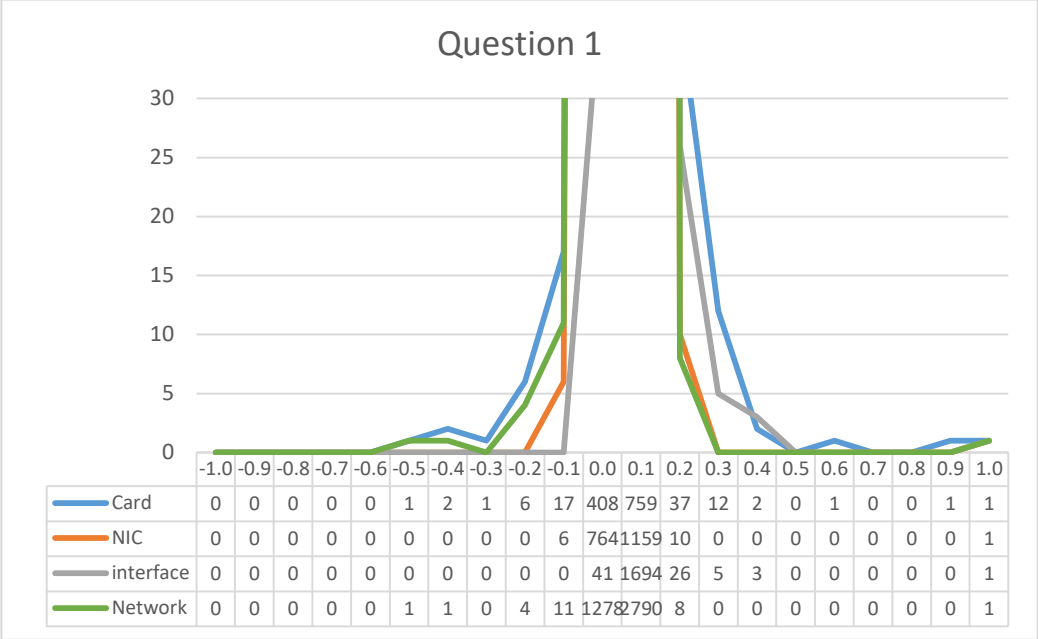
For the signature terms in Q1 the normal distribution frequency of the graph is shown below in Figure 4.2-1.



	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Card	0	0	0	0	0	1	2	1	6	17	408	759	37	12	2	0	1	0	0	1	1
NIC	0	0	0	0	0	0	0	0	0	6	764	1159	10	0	0	0	0	0	0	0	1
interface	0	0	0	0	0	0	0	0	0	0	41	1694	26	5	3	0	0	0	0	0	1
Network	0	0	0	0	0	1	1	0	4	11	1278	2790	8	0	0	0	0	0	0	0	1

Figure 4.2-1 - Question 1 normal distribution of weights

A more detailed graph displays the frequency distribution closer to the 95% boundaries is shown in Figure 4.2-2.



	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Card	0	0	0	0	0	1	2	1	6	17	408	759	37	12	2	0	1	0	0	1	1
NIC	0	0	0	0	0	0	0	0	0	6	764	1159	10	0	0	0	0	0	0	0	1
interface	0	0	0	0	0	0	0	0	0	0	41	1694	26	5	3	0	0	0	0	0	1
Network	0	0	0	0	0	1	1	0	4	11	1278	2790	8	0	0	0	0	0	0	0	1

Figure 4.2-2 - Question 1 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the signature terms are shown below. The topic term is on the first row where the signature terms are shown under each topic term. The signature terms that should not be included in the topic are crossed out and the bottom row shows the percentage precision.

Network	**Interface**	**NIC**	**Card**
ip routing address subnet devices internet broadcast	nic ~~arp~~ ~~collision~~ router bri interfaces bandwidth card	card adapter ~~ping~~ ~~collisions~~ pc fluke ~~category~~ ~~hubs~~ connector	nic pc board ~~internet~~
100% precision	75% precision	55.56% precision	75% precision

For the signature terms in Q2 the normal distribution frequency of the graph is shown in Figure 4.2-3.

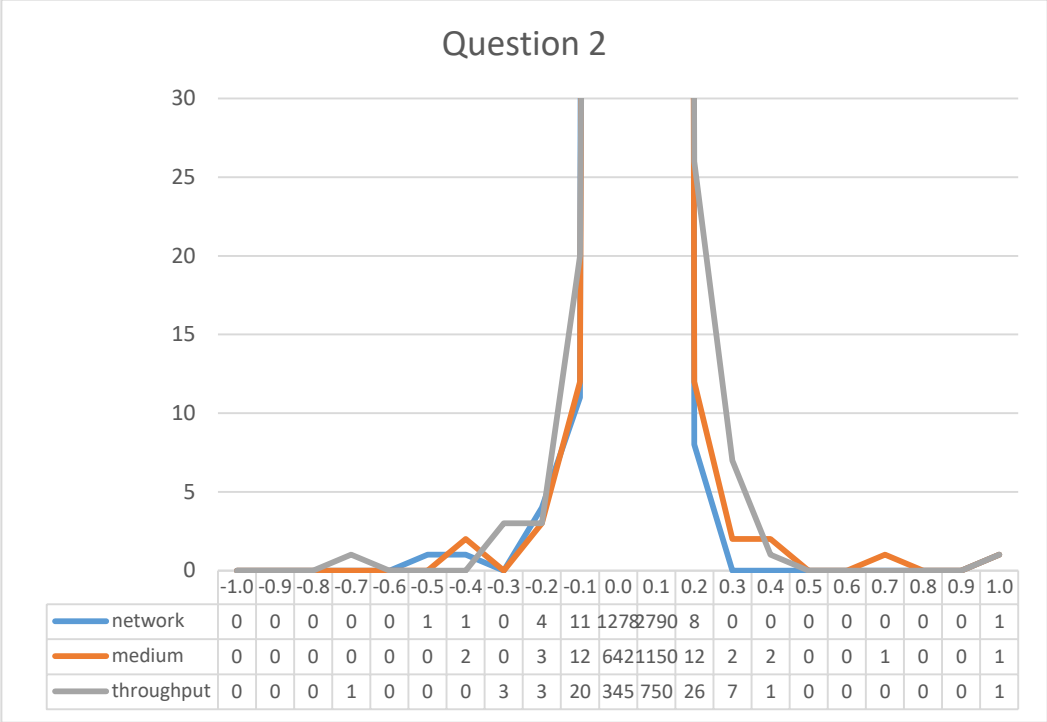


Figure 4.2-3 - Question 2 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

network	**medium**	**throughput**
ip	ethernet	bandwidth
routing	gigabit	~~802.11b~~
addresses	mbps	window
address	10gbe	jam
subnet		~~acknowledgment~~
devices		1000baset
internet		
broadcast		
100% precision	100% precision	66.67% precision

For the signature terms in Q3 the normal distribution frequency of the graph is shown in Figure 4.2-4.

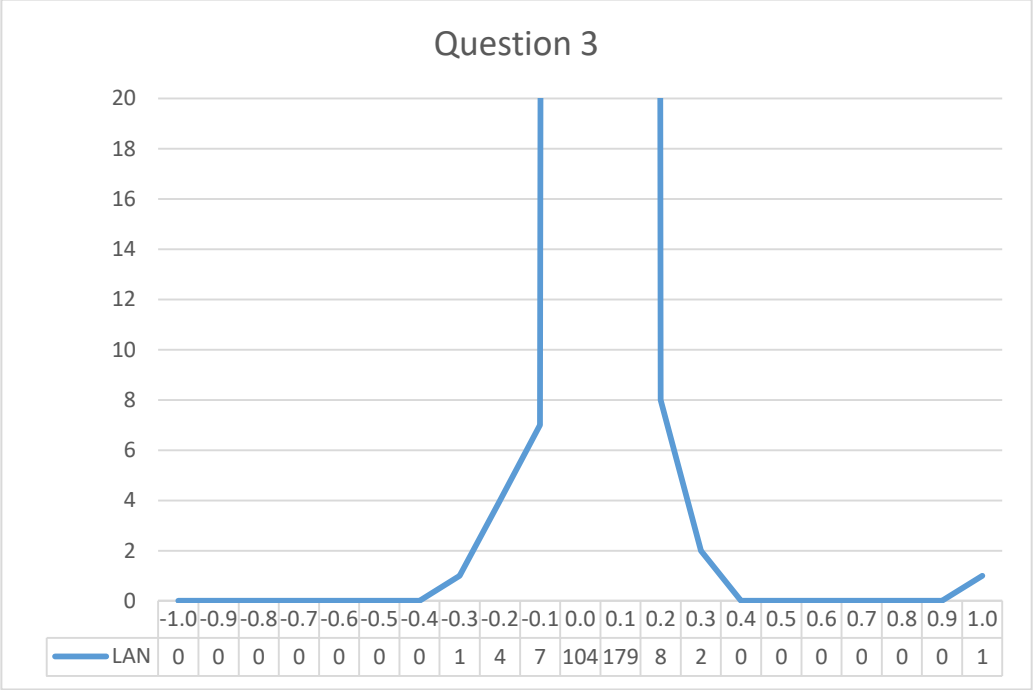


Figure 4.2-4 - Question 3 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

lan
devices arp wireless noise transmitter signals switches 802 area lans
100% precision

For the signature terms in Q4 the normal distribution frequency of the graph is shown in Figure 4.2-5.

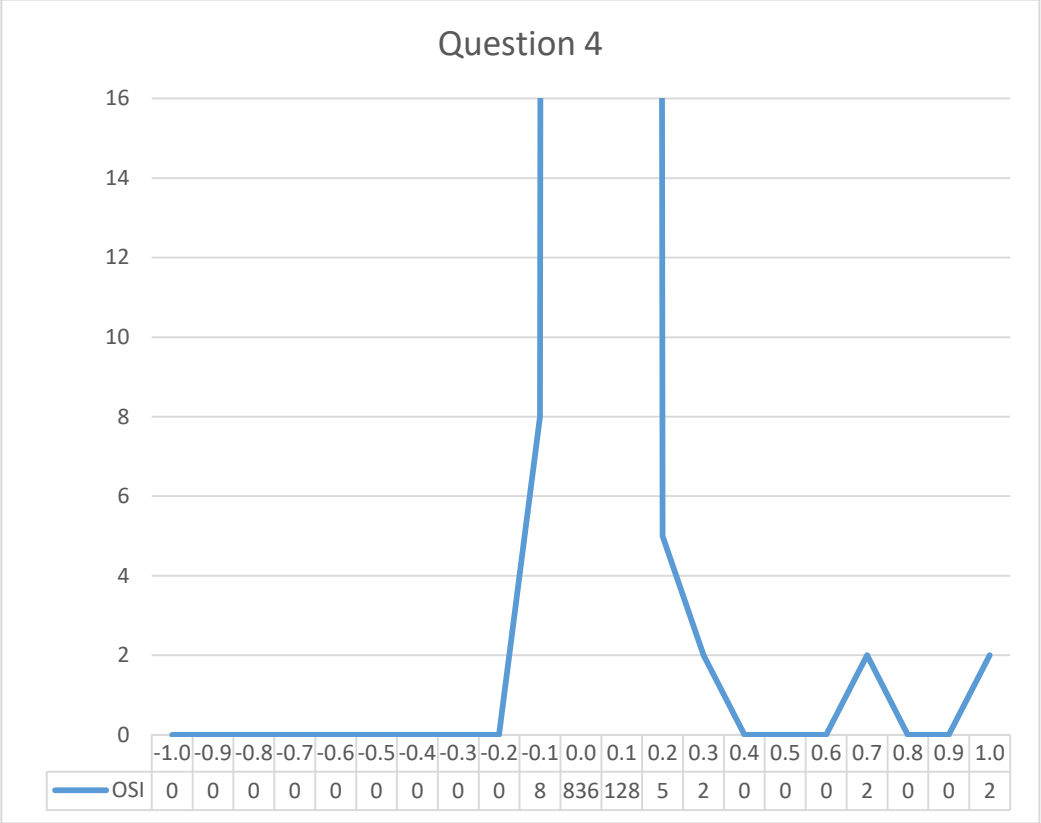
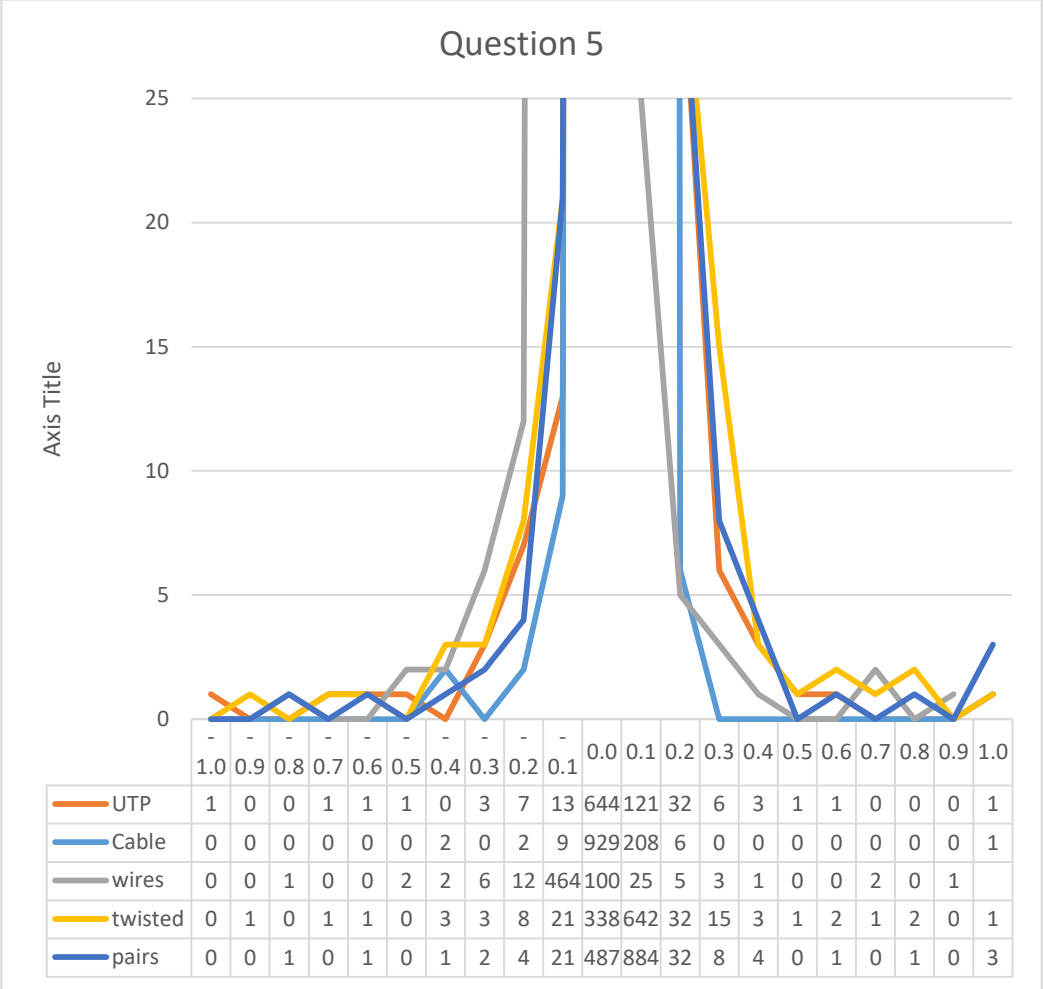


Figure 4.2-5- Question 4 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

OSI
model layer ~~tcp~~
66.67% precision

For the signature terms in Q5 the normal distribution frequency of the graph is shown in Figure 4.2-6.



	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
UTP	1	0	0	1	1	1	0	3	7	13	644	121	32	6	3	1	1	0	0	0	1
Cable	0	0	0	0	0	0	2	0	2	9	929	208	6	0	0	0	0	0	0	0	1
wires	0	0	1	0	0	2	2	6	12	464	100	25	5	3	1	0	0	2	0	1	
twisted	0	1	0	1	1	0	3	3	8	21	338	642	32	15	3	1	2	1	2	0	1
pairs	0	0	1	0	1	0	1	2	4	21	487	884	32	8	4	0	1	0	1	0	3

Figure 4.2-6 - Question 5 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

utp	cable	wires	twisted	pairs
cable wire pair 100basetx pairs	fiber noise wire category crosstalk connector	cable wire utp pair category structured crosstalk	cable wire stp pair utp sctp noise crosstalk shield shielded coaxial pins	cable wire pair crosstalk utp duplex noise category
100% precision	100% precision	100% precision	100% precision	100% precision

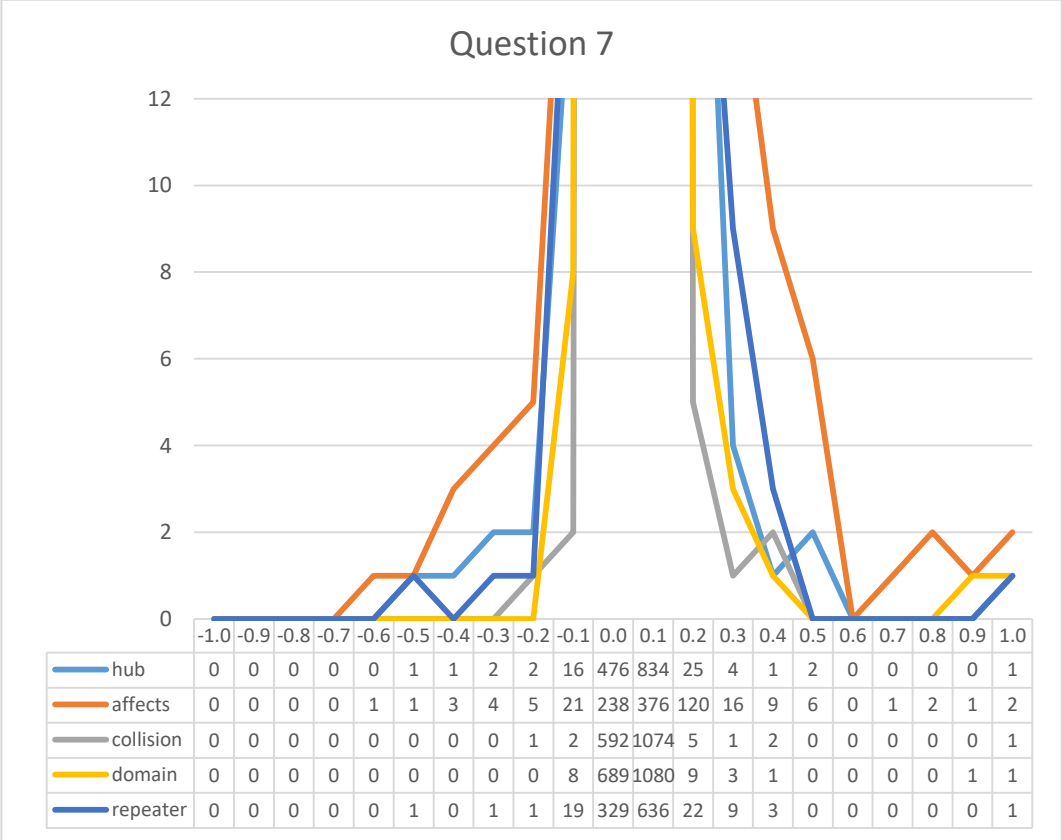
For the signature terms in Q6 the normal distribution frequency of the graph is shown in Figure 4.2-7.

Figure 4.2-7 - Question 6 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

electrons
resistance
current
charges
atoms
voltage
flow
protons
100% precision

For the signature terms in Q7 the normal distribution frequency of the graph is shown in Figure 4.2-8.



	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.3	-0.2	-0.1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
hub	0	0	0	0	0	1	1	2	2	16	476	834	25	4	1	2	0	0	0	0	1
affects	0	0	0	0	1	1	3	4	5	21	238	376	120	16	9	6	0	1	2	1	2
collision	0	0	0	0	0	0	0	0	1	2	592	1074	5	1	2	0	0	0	0	0	1
domain	0	0	0	0	0	0	0	0	0	8	689	1080	9	3	1	0	0	0	0	1	1
repeater	0	0	0	0	0	1	0	1	1	19	329	636	22	9	3	0	0	0	0	0	1

Figure 4.2-8 - Question 7 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

hub	affects	collision	domain	repeater
console 10baset ~~photozoom~~ rj-45 passive	narrowband noise broadcast broadcasts jam white multicast interference radiation organized block	domain frame station bridge broadcast	collision domains broadcast layer bridge	collision hubs
80% precision	not a signature term	100% precision	100% precision	100% precision

For the signature terms in Q8 the normal distribution frequency of the graph is shown in Figure 4.2-9.

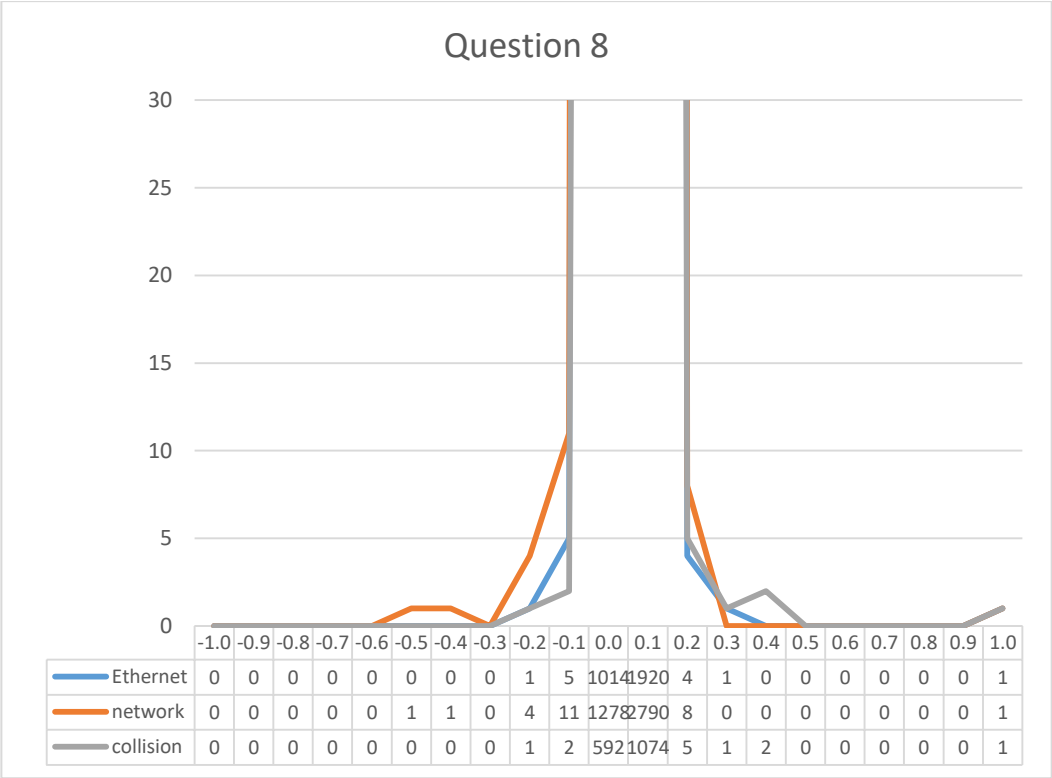


Figure 4.2-9 - Question 8 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

Ethernet	network	collision
collision gigabit frame ~~mbps~~ station	ip routing addresses address subnet devices internet broadcast	domain frame station bridge broadcast
80% precision	100% precision	100% precision

For the signature terms in Q9 the normal distribution frequency of the graph is shown in Figure 4.2-10

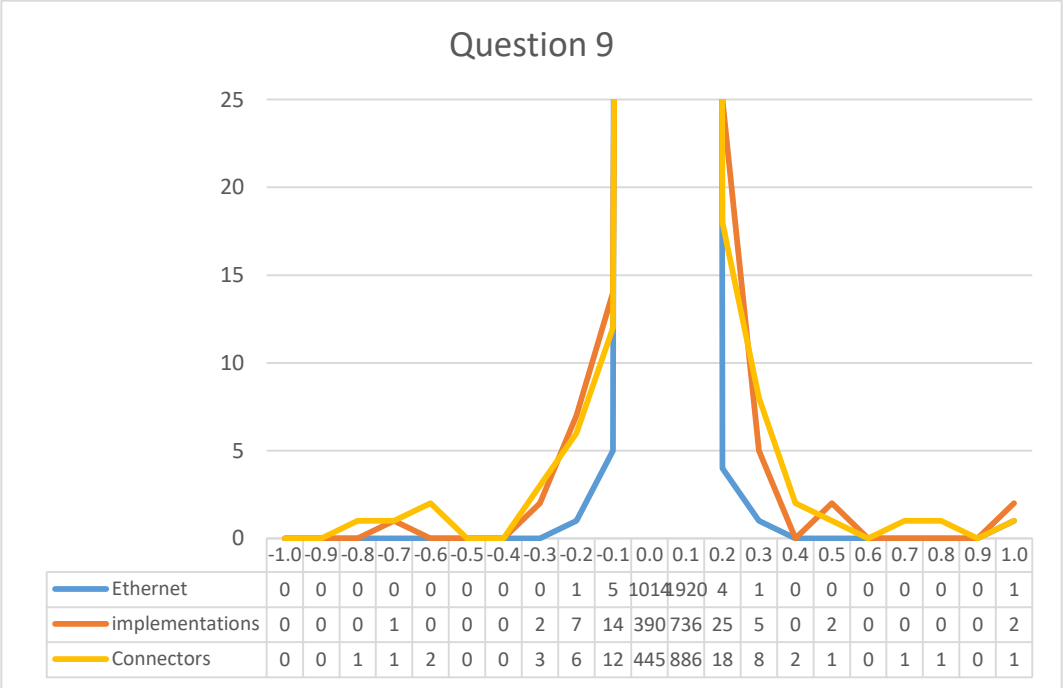


Figure 4.2-10 - Question 9 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

Ethernet	implementations	connectors
collision gigabit frame ~~mbps~~ station	ethernet duplex gigabit synchronous half ~~station~~ ~~timing~~ 10gbe	cable fiber rj crosstalk
80.00% precision	75% precision	100% precision

Finally, for the signature terms in Q10 the normal distribution frequency of the graph is shown in Figure 4.2-11.

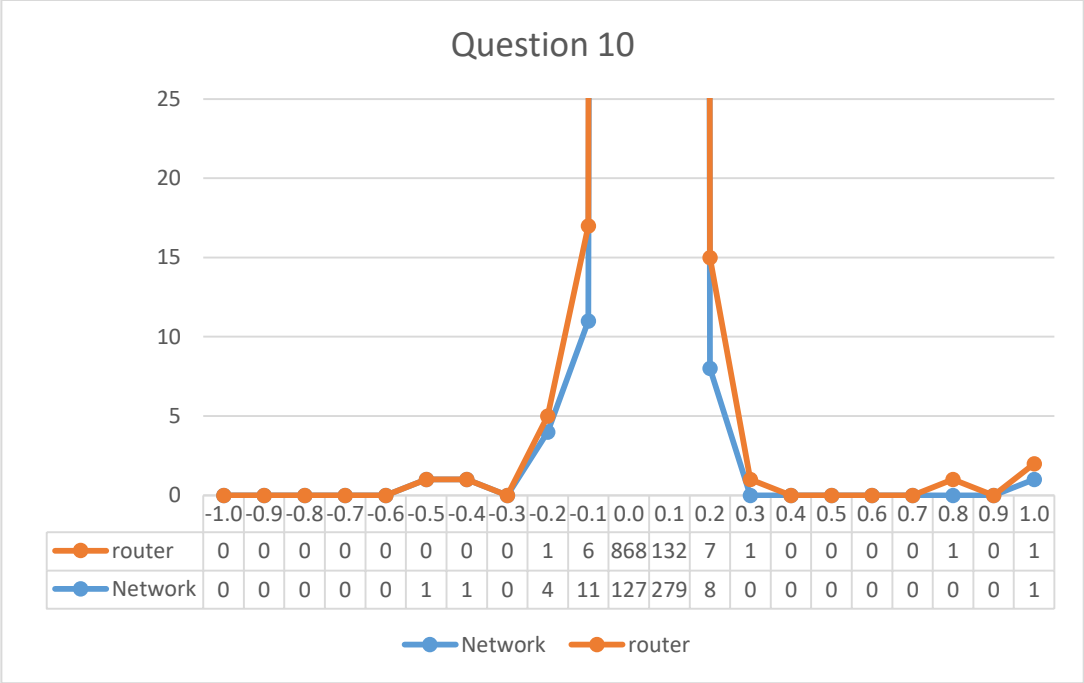


Figure 4.2-11 - Question 10 normal distribution of weights

The signature terms, which lay outside the 95% of the normal distribution for each of the topic terms are shown below.

router	network
routing address console straightthrough crossover linkstate metrics arp	ip routing address subnet devices internet broadcast
100% precision	100% precision

The next step was to evaluate the individual signatures extracted. The results shown in Table 4.2-6.

Table 4.2-6 – Topic Signature Evaluation

Topic	Signature	Total Signature terms	Correct Signature terms	Precision
cable	fiber, noise, wire, category, crosstalk, connector	6	6	100.00%
Card	nic, pc, board, internet	4	3	75.00%
collision	domain, frame, station, bridge, broadcast,	5	5	100.00%
connectors	cable, fiber, rj, crosstalk	4	4	100.00%
domain	collision, domains, broadcast, layer, bridge,	5	5	100.00%
electrons	resistance, current, charges, atoms, voltage, flow, protons,	7	7	100.00%
Ethernet	collision, gigabit, frame, mbps, station,	5	4	80.00%
hub	console, 10baset, photozoom, rj45, passive,	5	4	80.00%

Topic	Signature	Total Signature terms	Correct Signature terms	Precision
implementations	ethernet, duplex, gigabit, synchronous, half, station, timing, 10gbe	8	6	75.00%
Interface	nic, arp, collision, router, bri, interfaces, bandwidth, card	8	6	75.00%
lan	devices, arp, wireless, noise, transmitter, signals, switches, 802, area, lans	10	10	100.00%
medium	ethernet, gigabit, mbps, 10gbe	4	4	100.00%
network	ip, routing, address, subnet, devices, internet, broadcast,	7	7	100.00%
NIC	card, adapter, ping, collisions, pc, fluke, category, hubs, connector,	9	5	55.56%
OSI	model, layer, tcp,	3	2	66.67%
pairs	cable, wire, pair, crosstalk, utp, duplex, noise, category	8	8	100.00%
repeater	collision, hubs	2	2	100.00%
router	routing, address, console, straightthrough, crossover, linkstate, metrics, arp	8	8	100.00%
throughput	bandwidth, 802.11b, window, jam, acknowledgment, 1000baset	6	4	66.67%

Topic	Signature	Total Signature terms	Correct Signature terms	Precision
twisted	cable, wire, stp, pair, utp, sctp, noise, crosstalk, shield, shielded, coaxial, pins	12	12	100.00%
utp	cable, wire, pair, 100basetx, pairs,	5	5	100.00%
wires	cable, wire, utp, pair, category, structured, crosstalk,	7	7	100.00%
			Average	**89.72%**

From previous work on topic signatures (Lin, C., Hovy E., 2000) we can see that our method improves since human intervention is not required in this case and the precision of the algorithm is quite high.

4.2.2 Document Retrieval

In this section the evaluation of the module that retrieves the documents the answer will be extracted from will be presented. To begin this section, a brief explanation describes what the expected result should be. Then we give a breakdown of the different sets of results resulted from each run and explain them in detail.

4.2.2.1 Expected results

The expected results were derived from the CCNA self-assessment questions. The questions were picked in order to cover a large variety of question types and different answer types. The questions are based on a small subset of the CCNA learning material. This means that the answer provided by the self-assessment test is mapped with a document in the corpus that contains the text of the answer. The mapping is shown in Table

4.2-7. In questions 2, 7 and 9 two documents are mapped, since both documents contain the correct answer. If any of the documents is selected, then it will count as a success of the Question Answer system.

Table 4.2-7 - Correct answer /Document ID's

	Question	Document ID
1	Which describes the use of a network interface card (NIC),	814
2	Which is used to describe the rated throughput capacity of a given network medium?	845,849
3	What describes a LAN?	833,838
4	Why was the OSI model created?	854
5	Why are the pairs of wires twisted together in UTP cable?	901
6	What is required for electrons to flow?	866
7	How does using a hub or a repeater affects the size of collision domain?	961,963
8	Which will cause a collision on an Ethernet network?	961
9	Which Ethernet implementations use rj-45 connectors?	931,1041 1048
10	Which two functions of a router in a network?	1000

4.2.2.2 Base system results

In order to obtain some baseline results all the questions are passed through a lucence search engine. The first best answer was picked as the one the baseline system would return, which was not always the case when we run the system with the users (as demonstrated in section 4.2). Table 4.2-8 shows the answers as they were received from Lucence.

Table 4.2-8 - Lucene Answers

Question ID	Correct Answer	Lucene Answer	Correct
Q1	814	814	Y
Q2	845,849	845	Y
Q3	833,838	1016	N
Q4	854	854	Y
Q5	901	901	Y
Q6	866	866	Y

Question ID	Correct Answer	Lucene Answer	Correct
Q7	961,963	917	N
Q8	961	961	Y
Q9	931,1041 and 1048	915	N
Q10	1000	1000	Y

The baseline system (Lucene) scores very highly in terms of precision which is 70%.

4.2.2.3 Statistical Weights

In this section the results from the first run of the Question Answering task as described in 3.5.1 are presented. In this run, only the statistical weights, both TF.IDF and Log Likelihood and their sums and average scores are used. Only the learning object corpus is used which is referred to as the Domain Corpus. Table 4.2-9 – Domain Corpus document retrieval results shows the results of this run. If the document selected is the correct one, the cell has no shading and the text is bold. Incorrectly identified documents are shown in shaded cells with normal weight font.

Table 4.2-9 – Domain Corpus document retrieval results

	Domain Corpus (CCNA)			
	Sum Log Likelihood	**Average Log Likelihood**	**Sum TF.IDF**	**Average TF.IDF**
Q1	**814**	**814**	970	970
Q2	**845**	**845**	**845**	**845**
Q3	848	848	911	911
Q4	857	857	938	938
Q5	**901**	**901**	869	869
Q6	**866**	**866**	**866**	**866**

	Domain Corpus (CCNA)			
	Sum Log Likelihood	**Average Log Likelihood**	**Sum TF.IDF**	**Average TF.IDF**
Q7	**963**	**963**	**963**	**963**
Q8	**961**	**961**	**961**	**961**
Q9	943	1047	940	940
Q10	**1000**	**1000**	972	972

It can be seen that the Log Likelihood metric performs much better than the TF.IDF one. The TF.IDF metric has 40% precision in selecting the right document. On the other hand Log Likelihood without any other improvements, was just as good as the baseline results (70%). At this stage, it's worth mentioning that Lucene may not be a purely statistical information retrieval library, since it relies on some linguistic libraries.

Enabling the features described in sections 3.4.3 and 3.4.4 a new run of Document Retrieval is run. For this run only the Log Likelihood weights will be displayed since TF.IDF performed not as well as Log Likelihood did. The results are shown in Table 4.2-11

Table 4.2-10 shows the results obtained by running the same experiment against the Oxford corpus as described in section 3.5.2. The precision of picking the right answer is less at this run. Although it is important to mention that we wanted improvements to our algorithm, because the higher score in the results in Table 4.2-9 could be due to the smaller corpus size.

Enabling the features described in sections 3.4.3 and 3.4.4 a new run of Document Retrieval is run. For this run only the Log Likelihood weights will

be displayed since TF.IDF performed not as well as Log Likelihood did. The results are shown in Table 4.2-11

Table 4.2-10– Static Corpus document retrieval results

	Static Corpus			
	Sum Log Likelihood	**Average Log Likelihood**	**Sum TF.IDF**	**Average TF.IDF**
Q1	970	970	810	810
Q2	**845**	**845**	**845**	**845**
Q3	911	911	911	911
Q4	971	971	938	938
Q5	869	869	869	869
Q6	**866**	**866**	**866**	**866**
Q7	**963**	**963**	**963**	**963**
Q8	**961**	**961**	**961**	**961**
Q9	911	911	911	911
Q10	972	972	972	972

Comparing the answers returned by the two approaches, when the Oxford written English corpus is used the precision of picking the correct answer declined from 70% to 40%. This is an important decline in precision. After analysing the algorithm, it was identified that the decrease occurred because of the large size of the reference corpus.

This results show a correlation between the increase in the corpus size and the precision of document retrieval which decreases. The performance of

the two runs should at least be similar, since the size of the corpus used should not affect the performance.

Table 4.2-11 – Dynamic and Static corpus with Bigrams and term weights with stemming

Bigrams and term weights with stemming				
	Dynamic corpus		**Static Corpus**	
	Sum Log Likelihood	**Average Log Likelihood**	**Sum Log Likelihood**	**Average Log Likelihood**
Q1	**814**	**814**	**814**	**814**
Q2	**849**	**849**	**849**	**849**
Q3	919	919	931	931
Q4	**854**	**854**	**854**	**854**
Q5	**901**	**901**	**901**	**901**
Q6	**866**	**866**	**866**	**866**
Q7	917	917	917	917
Q8	883	883	883	883
Q9	913	913	**931**	**931**
Q10	**1000**	**1000**	1001	1001

Using stemming with bigram identification and query term weights enables the document retrieval module to perform almost as good as the baseline Lucene system which is a good achievement for the current state of the system.

The next step for improvement would be the use of topic signatures. If a topic and a signature term is present in the query, then the documents that contain the topic signature should be preferred for selection instead of other ones. Also if no signature term is present on the query, then documents that contain many topic terms or a high density of topic terms can be discarded. The reason behind this, is that if we think the document as an entity in vector space and the signature terms as boundaries of sub-topics

of the topic term, if the signature is present on the query then the required document would be within the signature space else it would be on a space where no signature terms are present.

A good demonstration for this scenario is question 3. This question contains one of the most general terms of the CISCO domain, the term *LAN*, without any signature terms. This causes the number of documents selected as a potential answer to be 107 which caused the wrong document to be picked up as the one that contains the answer. From the 107 documents, 52 contain more than one signature term. We discard these documents, since the ones with none or 1 signature term will contain more generalise information on the question. The documents remaining for selection are shown in Table 4.2-12

Table 4.2-12 - Filtered documents using Topic Signatures

Document ID	Signature Count	Document ID	Signature Count	Document ID	Signature Count
811	0	864	0	995	0
813	0	867	0	996	0
824	0	872	0	1007	1
832	0	873	0	1010	0
833	0	877	0	1011	0
835	0	883	0	1016	1
838	0	914	0	1017	1
839	0	922	0	1022	0
840	0	924	0	1033	0
844	0	928	0	1037	0
845	0	933	1	1038	1
848	0	942	1	1040	0
849	0	959	1	1041	1
850	0	966	1		
853	0	972	1		
856	0	975	1		
858	0	977	1		
859	0	983	0		

Document ID	Signature Count
861	0
862	0
863	0

Document ID	Signature Count
984	1
986	1
990	0

Document ID	Signature Count

The next feature to check would be the density of the topic term on the filtered documents from Table 4.2-12

Table 4.2-13 - Frequencies for Question 3 and Topic Signature "LAN"

Document ID	Topic Frequency
833	6
838	6
977	4
1010	4
835	3
859	3
933	3
832	2
839	2
840	2
844	2
858	2
861	2

For question 7 where the wrong document was picked, if we check the documents that were retrieved for the topic signature

Topic	collision
Signatures	domains frame domain station bridge broadcast stations broadcasts error frames jam legal

We get the following frequencies of the signature terms as shown in Table 4.2-14

Table 4.2-14 - Frequencies for Question 7 and Topic Signature "collision"

Document ID	Signature Frequency
857	2
888	4
963	42
967	39
1050	1

Again we can see that the expected document has the greatest density of topic signature terms. This can lead us to a conclusion that topic signature can improve the retrieval process of the Question Answering system significantly, if inappropriate machine learning algorithm is derived in order to consider the weight, any information from local analysis and the density from the topic signature analysis.

4.2.3 Answer pinpointing

In this section, the results of the answer pinpointing algorithm are shown.

4.2.3.1 Improvements by summarisation

The document retrieved for question 1 has an id of 814. Table 4.2-15 displays the document sentence ids together with the keyword frequency and if they are relevant to the question. The selected answer is highlighted in green with bold font (ids 19171-19177)

Table 4.2-15 – Q1 Answer Pinpointing

	Q1	
Document	814	
Stem	'network','interface','card','NIC','use','describe'	
Sentence ID	**Frequency**	**Relevant (Y/N)**
19165	3	N
19166	1	N

19167	0	N
19168	0	N
19169	0	N
19170	0	N
19171	**2**	y
19172	**2**	Y
19173	**1**	Y
19174	**1**	Y
19175	**2**	N
19176	**4**	Y
19177	**10**	Y
19178	0	

The text answer picked by the system is listed below with any irrelevant sentences highlighted. Each sentence is presented in a separate row.

A NIC must be installed for each device on a network.
A NIC provides a network interface for each host.
Different types of NICs are used for various device configurations.
Notebook computers may have a built-in interface or use a PCMCIA card.
Figure shows PCMCIA wired, wireless network cards, and a Universal Serial Bus (USB) Ethernet adapter.
Desktop systems may use an internal network adapter, called a NIC, or an external network adapter that connects to the network through a USB port.
Situations that require NIC installation include the following: Installation of a NIC on a PC that does not already have one Replacement of a malfunctioning or damaged NIC Upgrade from a 10-Mbps NIC to a

10/100/1000-Mbps NIC Change to a different type of NIC, such as wireless Installation of a secondary, or backup, NIC for network security reasons To perform the installation of a NIC or modem the following resources may be required: Knowledge of how the adapter, jumpers, and plug-and-play software are configured Availability of diagnostic tools Ability to resolve hardware resource conflicts The next page will describe the history of network connectivity.

In the sentences above there is only one which does not add information, but if media (images) were supported in the answer returned to the user, this sentence would also be relevant. The selected sentences have precision of 86%, while the ones ignored have a precision of 100%.

The document retrieved for question 21 has an id of 849.

Table 4.2-16 displays the document sentence ids with their keyword frequency and if they are relevant to the question. The selected answer is highlighted (ids 19748-19758)

Table 4.2-16 –Question 2 Answer pinpointing

	Q2	
Document	849	
Stem	'Network', 'use', 'capacity', 'throughput', 'medium', 'give','rate'	**Relevant (Y/N)**
Sentence PK1	**Stems frequency**	
19748	2	N
19749	1	Y
19750	1	Y
19751	1	Y
19752	0	Y
19753	2	Y
19754	2	Y

19755	7	Y
19756	2	Y
19757	4	Y
19758	1	Y
19759	0	N

The text answer picked by the system is displayed with any irrelevant sentences highlighted.

Bandwidth Throughput This page explains the concept of throughput.
Bandwidth is the measure of the amount of information that can move through the network in a given period of time.
Therefore, the amount of available bandwidth is a critical part of the specification of the network.
A typical LAN might be built to provide 100 Mbps to every desktop workstation, but this does not mean that each user is actually able to move 100 megabits of data through the network for every second of use.
This would be true only under the most ideal circumstances.
Throughput refers to actual measured bandwidth, at a specific time of day, using specific Internet routes, and while a specific set of data is transmitted on the network.
Unfortunately, for many reasons, throughput is often far less than the maximum possible digital bandwidth of the medium that is being used.

The following are some of the factors that determine throughput: Internetworking devices Type of data being transferred Network topology Number of users on the network User computer Server computer Power conditions The theoretical bandwidth of a network is an important consideration in network design, because the network bandwidth will never be greater than the limits imposed by the chosen media and networking technologies.
However, it is just as important for a network designer and administrator to consider the factors that may affect actual throughput.
By measuring throughput on a regular basis, a network administrator will be aware of changes in network performance and changes in the needs of network users.
The network can then be adjusted accordingly.

The selected sentences provide a full definition of the concepts and do cover the test question. Excluding the first sentence since it does not contain any information, our precision on picking the correct sentences is about 91% and the precision of excluding the unnecessary sentences is 100%.

In the case of Q3, the document picked by the Document Retrieval module is not the one that contains the correct answer. For this reason we pick document 838 which contains the correct answer because what is being tested at this phase is the ability of the module to extract the most relevant sentences.

Table 4.2-17–Question 3 Answer pinpointing

Q3		
Document	838	
Stem	1'LAN', 'describ'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
19605	2	N
19606	2	Y
19607	0	Y
19608	1	Y
19609	1	Y
19605	2	Y
19606	2	Y
19607	0	Y
19608	1	Y
19609	1	Y
19605	2	Y

The sentences selected by the system return are the following.

Networking Terminology Local-area networks (LANs) This page will explain the features and benefits of LANs.
LANs consist of the following components: Computers Network interface cards Peripheral devices Networking media Network devices LANs allow businesses to locally share computer files and printers efficiently and make internal communications possible.
A good example of this technology is e-mail.
LANs manage data, local communications, and computing equipment.
Some common LAN technologies include the following: Ethernet Token Ring FDDI The next page will introduce wide-area networks (WANs).

So we have 90.9% precision in picking the correct sentences.

Table 4.2-18 shows the breakdown per sentence and the frequencies of the keyword stems for document 854 which is the document that is contains the correct answer for question 4.

This question is a good example of how well the summarisation algorithm works in answer pinpointing. The precision reaches 100% and recall is also 100% in picking the correct 6 sentences out of a total of 20 sentences. For some of the sentences there is no presence of the keyword terms which makes it easier for the algorithm to work, but there are 2 main clusters in the document and the algorithm is capable of picking the correct one as the answer. If we compare this with returning the full document as per the baseline system, the precision achieved by Lucene would be 30% which underlines the improvement provided by our algorithm.

Table 4.2-18 –Question 4 Answer pinpointing

Q4		
Document	854	
Stem	'create','OSI','model'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
19844	5	N
19845	0	N
19846	0	N
19847	0	N
19848	0	N
19849	0	N
19850	0	N
19851	0	N
19852	0	N
19853	1	Y
19854	1	Y
19855	3	Y
19856	0	Y
19857	3	Y
19858	3	Y
19859	0	N
19860	0	N
19861	2	N
19862	2	N
19863	4	N

The text of the selected sentences is shown below

To address the problem of network incompatibility, the International Organization for Standardization (ISO) researched networking models like Digital Equipment Corporation net (DECnet), Systems Network Architecture (SNA), and TCP/IP in order to find a generally applicable set of rules for all networks.
Using this research, the ISO created a network model that helps vendors create networks that are compatible with other networks.

The Open System Interconnection (OSI) reference model released in 1984 was the descriptive network model that the ISO created.
It provided vendors with a set of standards that ensured greater compatibility and interoperability among various network technologies produced by companies around the world.
The OSI reference model has become the primary model for network communications.
Although there are other models in existence, most network vendors relate their products to the OSI reference model.

Table 4.2-19 displays the keyword frequencies corresponding to the sentences in the document that contains the answer for Q5 with the sentences picked by the algorithm highlighted in green, and the ones wrongly identified highlighted in red.

Table 4.2-19 –Question 5 Answer pinpointing

Q5		
Document	870	
Stem	'cable','wire','UTP','twist','pair'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
20322	2	N
20323	2	Y
20324	2	Y
20325	3	Y
20326	3	Y
20327	5	Y
20328	2	Y
20329	0	N
20330	0	N
20331	1	N
20332	0	N

Q5		
Document	870	
Stem	'cable','wire','UTP','twist','pair'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
20333	0	N
20334	0	N
20335	0	N
20336	1	N
20337	0	N
20338	1	N
20339	0	N
20340	2	N
20341	0	N
20342	1	N
20343	0	N
20344	2	N
20345	2	N
20346	0	N
20347	2	N
20348	0	N
20349	0	N
20350	0	N
20351	0	N
20352	0	N
20353	0	N
20354	0	N
20355	0	N
20356	0	N
20357	0	N
20358	0	N
20359	0	N
20360	0	N
20361	0	N
20362	0	N
20363	0	N
20364	0	N
20365	0	N
20366	0	N
20367	0	N
20368	0	N
20369	0	N
20370	0	N

Q5		
Document	870	
Stem	'cable','wire','UTP','twist','pair'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
20371	0	N
20372	0	N
20373	0	N
20374	0	N
20375	1	N
20376	2	N
20377	1	N
20378	1	N
20379	1	N
20380	0	N

The selected answer is shown below, with the irrelevant sentence highlighted in red.

Copper Media UTP cable This page provides detailed information about UTP cable.
UTP is a four-pair wire medium used in a variety of networks.
Each of the eight copper wires in the UTP cable is covered by insulating material.
In addition, each pair of wires is twisted around each other.
This type of cable relies on the cancellation effect produced by the twisted wire pairs to limit signal degradation caused by EMI and RFI.
To further reduce crosstalk between the pairs in UTP cable, the number of twists in the wire pairs varies.

Like STP cable, UTP cable must follow precise specifications as to how many twists or braids are permitted per foot of cable.

If the full document is returned to the learner, the precision of the answer pinpointing would be at 10% because there are many irrelevant sentences in the document. Keyword stems are scattered across the document which makes the selection process more difficult but the algorithm works well on that as well. The precision of the summarisation method is 87.5% with a recall score of 100%.

Table 4.2-20 displays the frequency sums of stemmed keywords for each sentence from the document that contains the correct answer for the sixth question. This is a somewhat special case, because the correct answer appears twice in the document. The way our algorithm worked in this case was to pick the cluster with the highest frequency density.

Table 4.2-20 –Question 6 Answer pinpointing

Q6		
Document	866	
Stem	'require','flow','electron'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
20183	0	N
20184	1	Y
20185	0	Y
20186	1	Y
20187	0	Y
20188	2	Y
20189	1	Y
20190	1	Y
20191	1	Y
20192	1	Y
20193	0	Y
20194	2	Y
20195	0	N

Q6		
Document	866	
Stem	'require','flow','electron'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
20196	0	N
20197	0	N
20198	0	N
20199	1	N
20200	1	N
20201	0	N
20202	1	N
20203	0	N
20204	0	N
20205	0	N
20206	1	N
20207	0	N
20208	0	N
20209	0	N
20210	1	N
20211	0	N
20212	2	N
20213	0	N
20214	0	N
20215	0	N
20216	1	N
20217	0	N
20218	0	N
20219	1	N
20220	0	N
20221	0	N
20222	0	N
20223	0	N
20224	0	N
20225	2	Y
20226	0	Y
20227	2	Y
20228	0	Y
20229	2	Y
20230	2	Y
20231	3	Y
20232	0	Y
20233	2	Y
20234	2	Y

Q6		
Document	866	
Stem	'require','flow','electron'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
20235	0	N
20236	0	N
20237	0	N
20238	0	N
20239	0	N

Depending which metric we use we arrive at different sentence clusters as the correct answer. Using the average stemmed keyword weight we would select the cluster that is formed from the sentences 20225 to sentence 20234. The answer of this method is shown below:

Electrons flow in closed circuits, or complete loops.
Figure shows a simple circuit.
The chemical processes in the battery cause charges to build up. This provides a voltage, or electrical pressure, that enables electrons to flow through various devices.
The lines represent a conductor, which is usually copper wire.
Think of a switch as two ends of a single wire that can be opened or broken to prevent the flow of electrons.
When the two ends are closed, fixed, or shorted, electrons are allowed to flow.
Finally, a light bulb provides resistance to the flow of electrons, which causes the electrons to release energy in the form of light.

If we use the length of cluster the cluster composed by using sentence 20184 to 20194 contains 11 sentences where the other one contains 10.

The first cluster will produce the following answer:

Current flows in closed loops called circuits.
These circuits must be made of conductive materials and must have sources of voltage.
Voltage causes current to flow.
Resistance and impedance oppose it.
Current consists of electrons that flow away from negative terminals and toward positive terminals.
These facts allow people to control the flow of current.
Electricity will naturally flow to the earth if there is a path.
Current also flows along the path of least resistance.
If a human body provides the path of least resistance, the current will flow through it.

In this case both answers are acceptable.

Moving to Question 7, the frequency of the stemmed keywords is shown in Table 4.2-21. For this evaluation the document that contains the correct answer was manually added as input to the answer pinpointing module. The correct answer is highlighted with green colour.

Table 4.2-21–Question 7 Answer pinpointing

	Q7	
Document	963	
Stem	'do', 'use', 'size',' repeater', 'hub', 'domain', 'affect', 'collision'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
22438	4	N
22439	1	N
22440	0	N
22441	0	N
22442	0	N
22443	1	Y
22444	0	Y
22445	1	Y
22446	0	Y
22447	1	Y
22448	0	N
22449	0	N
22450	0	N
22451	1	N
22452	0	N
22453	0	N
22454	1	N
22455	0	N
22456	0	N
22457	0	N
22458	0	N
22459	0	N
22460	0	N
22461	0	N
22462	0	N
22463	2	N
22464	0	N
22465	0	N

The answer produced by the algorithm is the following:

The types of devices that interconnect the media segments define collision domains.
These devices have been classified as OSI Layer 1, 2 or 3 devices.
Layer 2 and Layer 3 devices break up collision domains.
This process is also known as segmentation.
Layer 1 devices such as repeaters and hubs are mainly used to extend the Ethernet cable segments.

For Question 8 the sentences selected by the algorithm are shown in Table 4.2-22.

Table 4.2-22 –Question 8 Answer pinpointing

	Q8	
Document	917	
Stem	'network', 'Ethernet', 'caus', 'collis'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
21445	0	N
21446	0	N
21447	0	N
21448	0	N
21449	0	N
21450	2	N
21451	1	N
21452	1	N
21453	0	N
21454	0	N
21455	0	N
21456	0	N
21457	0	N
21458	0	N
21459	0	N
21460	0	N
21461	0	N

	Q8	
Document	917	
Stem	'network', 'Ethernet', 'caus', 'collis'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
21462	0	N
21463	0	N
21464	1	Y
21465	2	Y
21466	0	Y
21467	2	Y
21468	1	Y
21469	1	Y
21470	1	Y
21471	1	Y
21472	0	N

The text produced by returning the sentences selected by the algorithm is:

If many devices are attached to the hub, collisions are more likely to occur.
A collision occurs when two or more workstations send data over the network wire at the same time.
All data is corrupted when this occurs.
All devices that are connected to the same network segment are members of the same collision domain.
Sometimes hubs are called concentrators since they are central connection points for Ethernet LANs.
The Lab Activity will teach students about the price of different network components.
The next page discusses wireless networks.

The precision for this question is 71% with the two last sentences dropping precision due to the presence of the domain keyword "*network*".

For question 9 the results are shown in Table 4.2-23.

Table 4.2-23 –Question 9 Answer pinpointing

	Q9	
Document	913	
Stem	'Ethernet' , 'implement' 'Use', 'connector'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
21376	3	Y
21377	1	Y
21378	2	Y
21379	1	Y
21380	2	Y
21381	2	Y
21382	1	Y
21383	0	N
21384	0	N

The answer the system will provide according our algorithm is:

Cabling LANs Ethernet media and connector requirements This page provides important considerations for an Ethernet implementation.
These include the media and connector requirements and the level of network performance.
The cables and connector specifications used to support Ethernet implementations are derived from the EIA/TIA standards.
The categories of cabling defined for Ethernet are derived from the EIA/TIA-568 SP-2840 Commercial Building Telecommunications Wiring Standards.
Figure compares the cable and connector specifications for the most popular Ethernet implementations.

It is important to note the difference in the media used for 10-Mbps Ethernet versus 100-Mbps Ethernet.
Networks with a combination of 10- and 100-Mbps traffic use Category 5 UTP to support Fast Ethernet.

In this case the system achieves 100% precision and 100% recall.

Finally, for question 10 the stemmed keyword frequencies are shown in Table 4.2-24.

Table 4.2-24 –Question 10 Answer pinpointing

	Q10	
Document	1000	
Stem	'router' , 'function'	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
23395	2	N
23396	1	N
23397	0	N
23398	0	N
23399	0	N
23400	1	Y
23401	4	Y
23402	1	Y
23403	1	Y
23404	1	Y
23405	1	Y
23406	0	N
23407	0	N
23408	1	Y
23409	1	Y
23410	1	Y
23411	1	Y
23412	1	Y
23413	1	Y
23414	0	N
23415	0	N
23416	0	N
23417	0	N
23418	0	N
23419	0	N
23420	0	N

	Q10	
Document	1000	
Stem	‘router’ , ‘function’	
Sentence PK1	**Stems frequency**	**Relevant (Y/N)**
23421	0	N
23422	0	N
23423	0	N
23424	0	N

Using the longest sentence sequence metric the answer we return to the learner would be:

Routers interconnect network segments or entire networks.
Routers pass data frames between networks based on Layer 3 information.
Routers make logical decisions about the best path for the delivery of data.
Routers then direct packets to the appropriate output port to be encapsulated for transmission.
Stages of the encapsulation and de-encapsulation process occur each time a packet transfers through a router.
The router must de-encapsulate the Layer 2 data frame to access and examine the Layer 3 address.

4.3 User Evaluation

In this section the results from the user evaluation are presented. This will support hypothesis H5 (*Using the Information Extraction techniques, students will be able to get the correct answer quicker and looking in fewer places than using standard search engines*). For this experiment, as described in the methodology section, a group of MSc Computing students used an interface to the system. This interface gave access to the actual corpus we used for the QA system and relied on a widely used java library (Lucence) in order to search through the document collection. The results of the search were displayed in a similar way as Google displays the results of a web search, showing the first few lines of text. For more details please refer to the methodology section. The student's answers are grouped per document and the percentage of times a document being picked by a student is calculated. If a question has one or more green cells it means that this document contains the answer.

From Table 4.3-1 it is visible that there are questions for which the document containing the correct answer was not picked by users. Comparing the results, with the correct answers that can be retrieved from our Question Answering system, we can see that there is a clear advantage on using the Question Answering system, since the student will have overall more correct answers available.

Table 4.3-1 – User selected documents

Question	Document ID	Selected from	%
1	814	2	33.33
	813	4	66.67
2	849	5	83.33
	845	1	16.67
3	837	1	16.67
	911	2	33.33
	919	1	16.67
	1037	1	16.67
	952	1	16.67
4	832	1	16.67
	854	4	66.67
	969	1	16.67
5	903	2	33.33
	902	1	16.67
	901	2	33.33
	870	1	16.67
6	862	3	50.00
	866	3	50.00
7	945	1	16.67
	963	2	33.33
	917	1	16.67
	961	1	16.67
	916	1	16.67
8	883	1	16.67
	945	1	16.67
	947	1	16.67
	962	1	16.67
	946	1	16.67
	960	1	16.67
9	914	4	66.67
	915	1	16.67
	931	1	16.67
10	927	1	16.67
	835	1	16.67
	1000	4	66.67

A breakdown of correct answers per user is shown in Table 4.3-2

Table 4.3-2 – Correct answers per user

StudentID	Number of Correct Answers
student1	2
student2	4
student3	5
student4	5
student5	4
student6	8
Average	**4.67**

The maximum correct answers that a student could pick up using the search engine were 8 questions, but the deviation between the scores is fairly large, with student 1 only being able to identify two right answers. Student 6 has the best score which is 8 right answers. The average user, based on our sample is expected to have 4-5 right answers which is fairly poor considering that a search engine can be used to help student progress their learning.

The average score of correct answers identified by the user can be achieved by the Question Answering system only using the statistics measures. Without the Bigram identifications, Topic signatures and Query terms weights produced 6 correct answers, which is above the average the students have selected using the search engine.

Using the final version of the QA system (with Bi-gram detection, Query Term Weighing and Topic Signatures), the students will be presented with more correct answers than if they used a search engine to retrieve the answer. The issue with such anapproach is that the student will not be presented with other answers which may contain the correct answer. For this reason, we can provide the student with the top 5 answers and they would be able to navigate to different answers as a possible enhancement to the system.

Another metric calculated in the user evaluation is the number of clicks per question that a user makes when looking for an answer. After a search is completed the results are shown on the screen with the document title only appearing in a similar way as Google displays the search results. Each time

the user wants to see the full content of the document they have to click on the title of the document. They can then select the text as the correct answer, and the total amount of clicks is stored in the database.

The figure below shows a breakdown of clicks per question per user. The Y axis has the number of clicks a user has made from starting the question till they select a document as relevant. The X axis is mainly grouped by student and each bar represents the number of clicks made by a user to retrieve the answer to one question. Each question is identified by the index of the question starting from 1 and the last question being number 10.

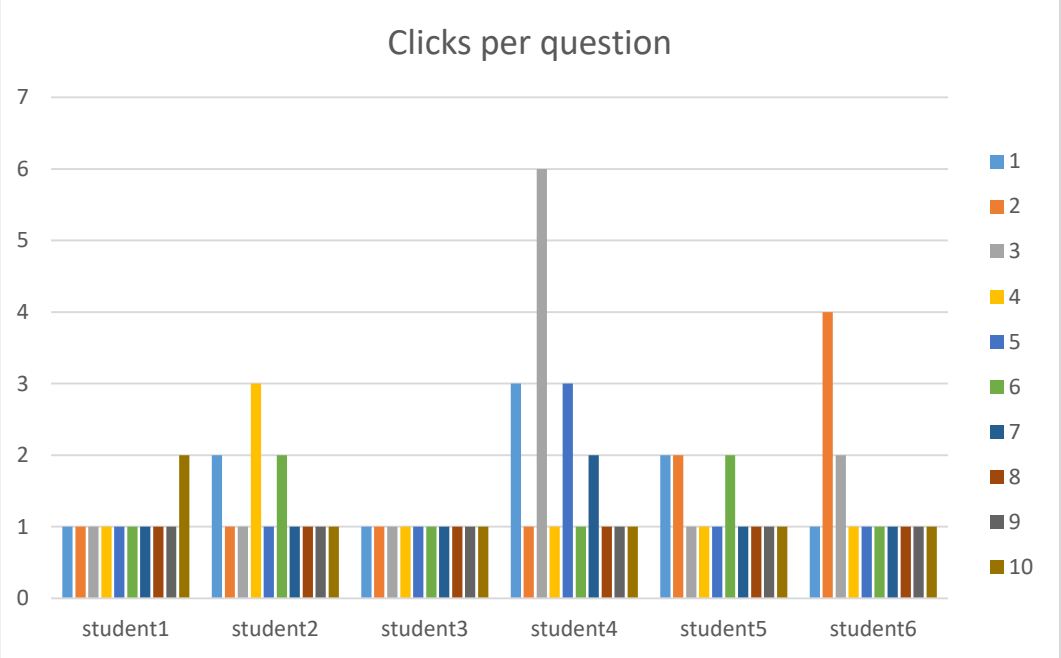


Figure 4.3-1 – Baseline system clicks per question

If we compare Figure 4.3-1 with Table 4.3-2, it is obvious that there is no correlation between the amount of documents opened by the user and the selection of the correct answer. For example *student3* picked the first

document for all questions while *student4* always looked into more documents to find the correct answer. The number of correct answers picked by the user were the same in both cases. So when a student ask the QA system a question and the answer returned by the system is the correct one, the student will not be viewing information irrelevant to the query.

From Figure 4.3-1, it's clear that only one student was able to pick the answer from the documents they chose the first time. Some students were more confident in picking the answer like *student1* and others explored more answers before picking an answer such as student4. In comparison with the search engine, when using the QA system the user would not need to click on different documents in order to retrieve the correct answer. As we can see *student1* who opened the least amount of documents, had the smallest amount of correct documents picked. *Student4*, who opened most documents had picked just over the average amount of correct answers. *Student6* on the other hand with an average amount of opened documents has identified the highest amount of correct answers.

The next metric examined was the number of searches the user performed per question. This metric is mainly captured to confirm that the students searched multiple times to answer a question. The reason for this is to emphasise the usefulness of the QA system which will always give one answer for a question, without requiring multiple searches. Evaluating the QA system against the baseline search engine, we can also see that the QA system can be faster than the baseline system. If the answers from both systems are correct and the user triggered more queries in order to get the

right answer, then our QA system is quicker in response and would evaluate better. Figure 4.3-2 shows the number of searches per question. On the Y axis we have the amount of searches. The X axis is divided per user and each bar corresponds to a question per user. The questions are labelled with their index in the system starting with 0 for the first question and index number 10 for the last question.

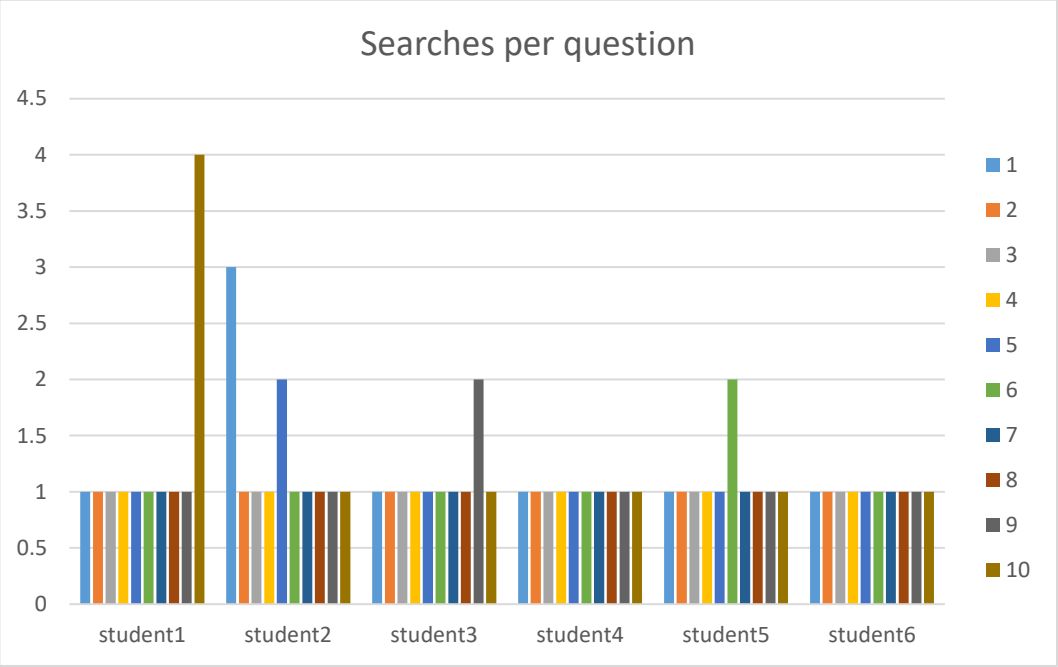


Figure 4.3-2 – Baseline system searches per question

The next interesting metric is the time that each student spent on each question. This metric is important as spending more time on a question than average means that there is a difficulty in finding the correct answer

The users are MSc Computing students who have been taught basic notions of networking, so their judgement is considered to be better than that of the novice users. Figure 4.3-3 displays the time each student spent on a question, there is no desirable trend in order to identify any harder or simpler questions or questions which were not too hard and did not require

any critical analysis from the students. There is no normal distribution on the time spent on each question, which can be perceived as there being no question that is harder than the other questions, otherwise all the students would have spent more time on that particular question.

This enforces the assumption that using a search engine can be too difficult from a student's perspective. Compared with our QA system, the answer would be immediate and also the answer is more query targeted.

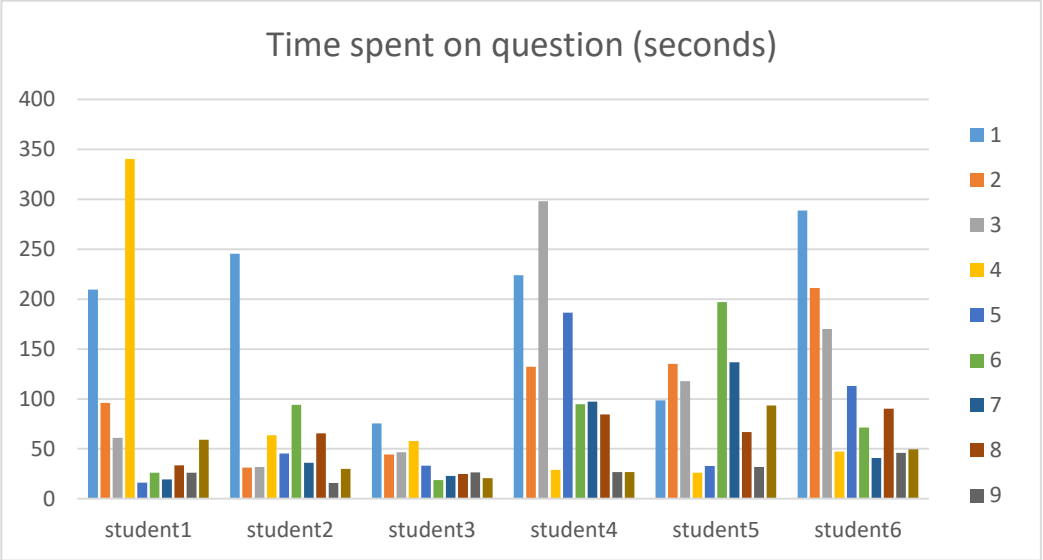


Figure 4.3-3 – Baseline system time spent on questions

To interpret the above figure a little differently, the average time spent per question is shown in Figure 4.3-4. The time to retrieve the response from the server is usually in the region of a couple of seconds as we can see from Figure 4.3-5. So the student can spend longer trying to find what they believe is the correct answer from a document collection. Although looking for information is part of the learning journey of the student, spending time to retrieve information not directly connected to a task can be time consuming and divert the student's attention.

InFigure 4.3-6 – Time spent per question using the QA system vs Baseline search engine., we can see the time each student spent on each question using both systems. The QA system supported the student quicker than the search engine, for example Question 1 and Question 4 for student 1.

The Question Answering system described in this thesis, when compared to the traditional searching approach, can respond to the user faster every time and significantly faster overall when all the questions are considered.

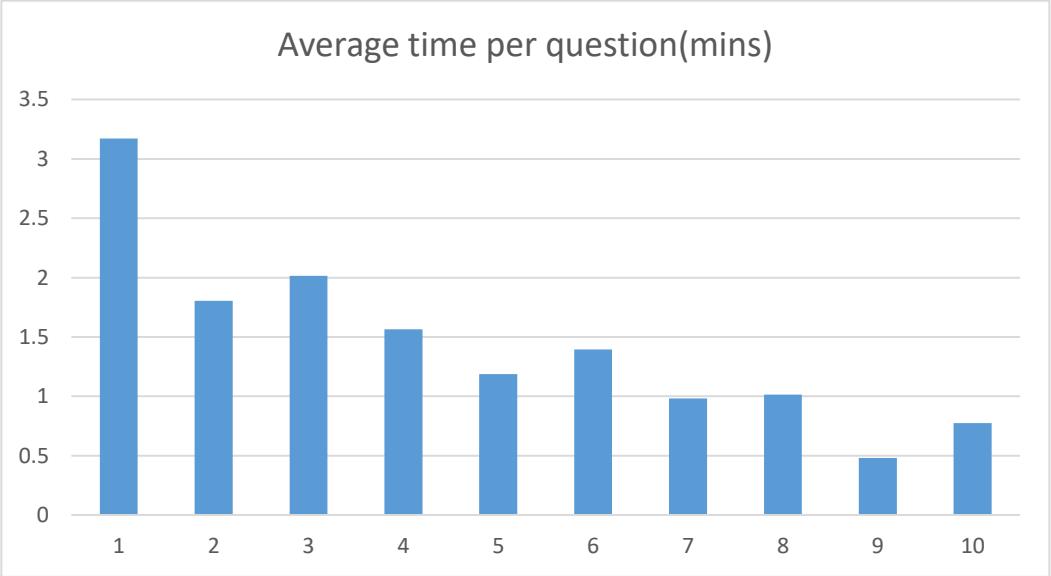


Figure 4.3-4 - Average time per question

URL	Status	Domain	Size	Remote IP	
Net panel activated. Any requests while the net panel is inactive are not shown.					
GET search?term=which%20of%2..	200 OK	buddie-devsol.rhcloud.com	31.6 KB	54.211.155.0:80	2.53s
GET search?term=which%20of%2..	200 OK	buddie-devsol.rhcloud.com	31.6 KB	54.211.155.0:80	1.9s
GET search?term=which%20of%2..	200 OK	buddie-devsol.rhcloud.com	31.2 KB	54.211.155.0:80	1.73s
GET search?term=which%20of%2..	200 OK	buddie-devsol.rhcloud.com	31.3 KB	54.211.155.0:80	1.64s
GET search?term=router%20issues	200 OK	buddie-devsol.rhcloud.com	21.7 KB	54.211.155.0:80	1.14s
GET search?term=LANS	200 OK	buddie-devsol.rhcloud.com	29.7 KB	54.211.155.0:80	1.29s
8 requests			239.8 KB		21.74s

Figure 4.3-5 – Question Answering response times

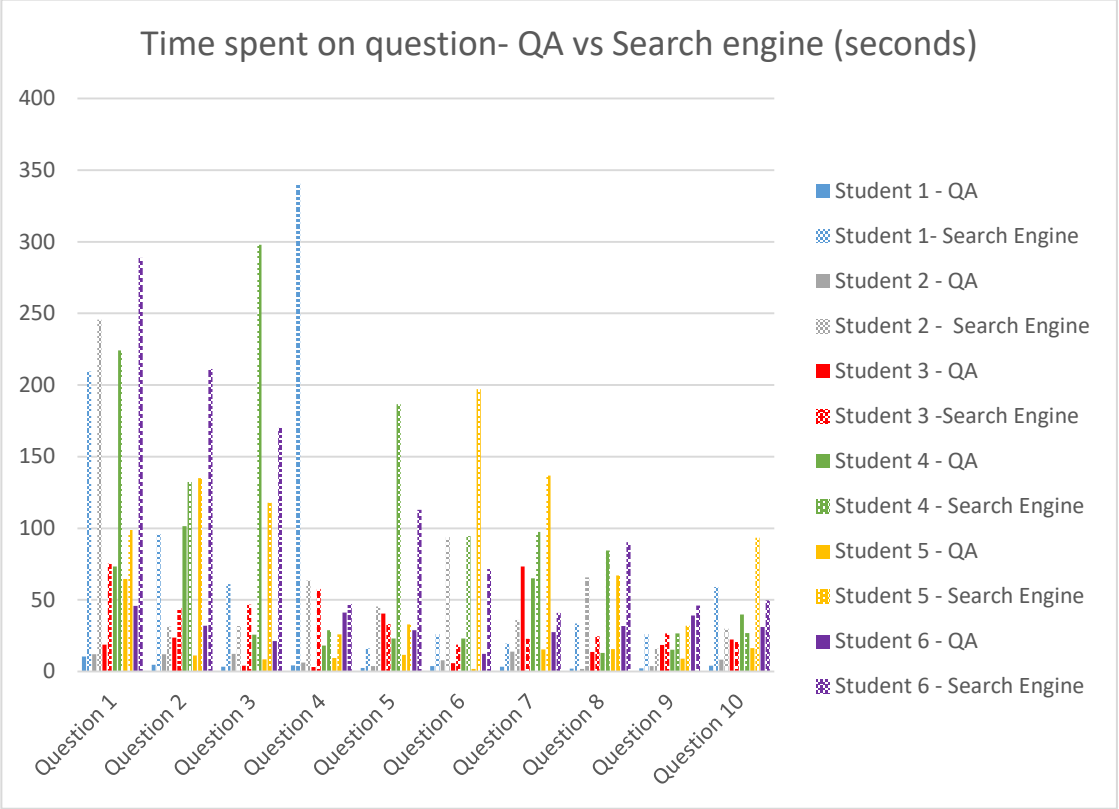


Figure 4.3-6 – Time spent per question using the QA system vs Baseline search engine.

A summary of the time difference between the QA system and the baseline search engine is shown in Table 4.3-3.

Table 4.3-3 - QA vs Baseline time difference

	Question									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
student1	199.03	91.25	57.61	336.49	13.79	22.33	16.18	31.59	23.89	54.91
student2	233.39	19.30	19.50	57.19	41.61	86.10	22.24	63.95	12.13	21.80
student3	56.60	20.63	42.63	54.53	-7.35	12.99	-50.60	11.10	7.85	-1.88
student4	150.94	30.82	272.43	10.65	163.44	71.54	32.29	71.54	11.58	-12.97
student5	34.19	123.98	109.18	16.55	21.22	195.48	121.41	51.35	22.94	77.08
student6	243.08	179.27	148.79	6.05	84.12	59.26	13.51	58.40	6.90	18.55

Finally after each question the users needed to point out which answer was their preferred answer after they used both systems. The results are shown in Table 4.3-4 - User system preference, with each row and column having

an average preference to the QA calculated. The majority of students preferred using the QA system over the Search interface. Even in a situation where the student had the same answer from both systems, he said that he preferred the QA because he did not have to look into the documents for the information.

Table 4.3-4 - User system preference

Students	**Question indexes**										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**Preference of QA**
student1	Search	Search	QA	QA	Search	QA	QA	QA	QA	QA	70.00
student2	QA	QA	QA	QA	QA	QA	QA	QA	QA	QA	100.00
student3	QA	QA	QA	QA	QA	QA	QA	QA	QA	QA	100.00
student4	QA	QA	QA	QA	QA	QA	QA	QA	QA	QA	100.00
student5	QA	QA	QA	QA	QA	QA	QA	QA	QA	QA	100.00
student6	QA	QA	QA	QA	QA	QA	QA	QA	QA	QA	100.00
Preference of QA	83.33	83.33	100.00	100.00	83.33	100.00	100.00	100.00	100.00	100.00	

At the end of the evaluation some extra information was gathered about the user experience and also students could add a comment in a textbox regarding anything else they wanted to say. The results of the final feedback form the students is shown below in Table 4.3-5. For each student we have the results for the simple Yes/No answers and in the row below the comment the student put in the text box. One out the six students did not find the system useful for their study. All the other students could see the potential of a QA System in their VLE. Something that can be looked into as future research, is to measure the effectiveness of a QA systems with respect to different learning styles. Also only one student did not find the immediate feedback essential. Their main reason, was that although the system supported the students quickly, the actual learning process of

the learner is different. Learners may need to look into the content and investigate a topic to gain a deep understanding. How to use a QA tool effectively in a learning environment is out of the scope of this research where we wanted to investigate if a QA system can perform better than the existing baseline search techniques. Having this hypothesis evaluated and by proving that there is space for QA systems in leaning environments, opens a whole new area of research as to how these tools should be used. The main scenario underlying the development and testing of the system was for the QA system to be used as a support tool and not as a learning tool. The main difference is that the students should be able to retrieve important query driven snippets of information when they need to or while working on an assignment. Another important point raised by the users is that they want quality materials online. Although our system only contained CCNA approved materials, as a general comment the students wanted quality resources available to them via the QA interface. This can easily be accomplished since the QA system will only work on resources their lecturers have uploaded on their VLE.

Table 4.3-5 – User Feedback

Student ID	Can the QA system aid your study	Is immediate learning support very important	Did you find the QA system useful
student1	no	yes	yes
	Comment:		
student2	yes	yes	yes
	Comment: it is necessary to have a local and internet dependent knowledge repository where I can be sure the information is of a trusted source		
student3	yes	yes	yes
	Comment:		
student4	yes	no	yes
	Comment: Being told the right answers is good in time constrained situations but actual learning is different and effective only when you go through the wrong and right answers before deciding which the best is.		
student5	yes	yes	yes
	Comment: Immediately learning support saves your time and efforts		
student6	yes	yes	yes
	Comment: Any reliable concise source of information that is 'immediate' will be a great help. Time is limited on MSc programmes so a learning aid like this would be very useful at the start of a new module.		

Chapter 5

5 Conclusion

5.1 Introduction

As described in Chapter 2, Question Answering systems have not been widely used in Virtual Learning Environments. Some attempts have been made from time to time but they did not become a mainstream support tool. The issues with current systems come from the need to have an expert either to create a knowledge base (entities, ontologies etc.) for the content to be used by a Question Answering system which is hard to maintain due to the amount of data that is continuously uploaded to the Virtual Leaning Environment.

Question answering systems can provide very good support to learners for a range of content. As long as the answer is present in a document, the system can provide it to all the students at any time they request it which is a great advancement in the area of automatic student support. Having no teaching staff input to the system makes the solution portable and also inexpensive to run. In general Question Answering systems have been researched since early 1970 with systems such as LUNAR. This makes them a mature area of research where many different approaches have been evaluated and morphologically different types of corpora have been used. The domains Question Answering system have been applied to be numerous, but within a Virtual Learning Environment the systems that have been developed are rare. The technological approach that is taken by most of the system falls into two main categories: the statistical ones, and the Natural Language Processing ones (taggers, syntactic parsers, etc.). The statistical approach in information retrieval is not a new feature either. Pioneering approaches came from J. Firth in 1957.

In this thesis, the work undertaken to answer self-assessment questions, available in the CCNA online student notes is described. In order to obtain the answer, the only tools used are statistical methods, and in some parts of the system, some static lists and sentence splitters have been used, which may be considered as NLP (Natural Language Processing) approaches, but can be replaced by equivalent statistical methods. The principal aim of this research was to use and enhance a successful combination of the techniques used in the Question Answering task to provide 24/7 quality support to a student at a more advanced level than existing support tools do. By developing a portable Question Answering system, new opportunities are created in the area of user support, so the learner has an improved learning experience, by helping the student locate information faster, with less irrelevant data fed back to the student. From our user evaluation, it's understandable that since the student is not an expert user, they can be easily mislead by the content returned from a search engine and select the wrong answer.

The evolution of browser based frameworks and the architecture we used to develop the services used to answer questions, other applications can benefit from such tools like dialogue systems, active content generation etc.

5.2 Contribution to current knowledge

In this section we summarise the main contributions of this research project.

5.2.1 Statistical methods for answering questions in a VLE

5.2.1.1 Originality

The first aspect of originality of this research work comes from developing an algorithm that uses statistical methods in order to answer user questions within a VLE. Our question answering system as explained in chapter 3, comprises different submodules. The query processing module uses techniques described in section 2.3.1.1. Examples of the usage of local techniques are in other systems are presented in section 2.3.1. In our implementation local analysis was used slightly different. Each time some background knowledge was required, local techniques were applied. In previous literature, local techniques were mainly used for query parsing. The Document Retrieval module, is a more advanced algorithm than the solution used to retrieve documents by Fisher, Roark (2006). For our corpus, we evaluated the performance of the algorithm with and without stemming and stemming in our algorithm produced better results. An enhancement to the existing techniques was to calculate the document weight based on the information passed from the Query parsing module. We based the score of a document, on the weight of each of the query terms in the corpus in relation with the weight the term has using local analysis. This produced better results than using only the term weights derived from the corpus. Also documents that were weighted higher because of noise created by less important terms in the query, in cases where a less important term appeared multiple times and skewed the results by disfavouring more important terms, by using this enhancement were weighted more fairly.

Our algorithm also works on smaller and larger corpora. This is a part of the evaluation of the hypothesis H2, where we want so ensure that the algorithm will not be biased on the size of the corpus. As evaluated in section 4.2.2, the same amount of documents were retrieved when we used one learning object as the reference corpus and when we used the Oxford corpus. Statistical methods can be biased on the corpus size and generally their performance varies with respect to the corpus size. The combination of technologies selected, after conducting the experiment supports the hypothesis that the algorithm developed is independent of the size of the corpus.

5.2.1.2 Hypothesis Review

H1. The correct answer to a question entered to the system should be retrieved using statistical methods and without requiring any background knowledge.

H2. The statistical approaches used should not be dependent on the size of the corpus

H3. A good combination of methods that will work on a learning domain to answer user specific questions are:

a. Log Likelihood

b. Summarisation techniques - to extract sentences relevant to the user query

The first hypothesis stated that there should be a successful combination of statistical technologies that can be used in a learning environment in order to retrieve the correct answer using only the content available in the Learning Environment. This hypothesis was supported in many of the

evaluation methods, by module and overall system design. Using as a baseline system a search engine, it is proved that using the algorithms developed the same amount of correct answers as a baseline search engine can retrieve, but the quantity of answers is higher since only the correct passages are returned to the student. Also there is no guarantee that the student will always select the top answer. From the search engine the first result that was returned and checked if the correct answer was present in the document. The comparison of the two system can be found in sections 4.2.2.1(search engine) and 4.2.2.2(QA system). The results in 4.2.3 show that the algorithm performs better than the baseline system, since the sentences returned to the user are shorter than the ones returned from the baseline system.

Also comparing with a realistic scenario, where students pick the answers that are listed by the search engine in weighted order, instead of picking the first answer that was returned by the system, our system will perform more than twice as well as the baseline system. The average correct answers selected by students were 4.6 out of 10 while the number of correct answers picked up by our system at the final stage were 8 out of 10. This clearly supports our hypothesis about being able to answer student questions using only the corpus and statistical weights.

We can also see that our system satisfies hypothesis 2, where the algorithm developed is independent of the size of the corpus. This is a major finding and the need to find support to this hypothesis is crucial. During the first runs of the development we could see that when we were running the statistical test on our datasets, in the test that an external corpus was used

the results had produced some noise and the incorrect document scored higher than the correct one. As explained in section 3.5.2.2 we overcome this issue using local techniques and the results in 4.2.2.3 demonstrate that.

To recap, so far we managed to support our hypotheses that having a statistically based algorithm in a Virtual Learning environment that returns the correct answer to student questions and also performs the similarly irrelevant of the size of the corpus. During our research and development we developed another hypothesis H3 where we proved that log likelihood is a much better measure than TF.IDF. However in this domain there was not much difference in the sum or average of the log likelihood measure. This was largely expected since log likelihood performs generally well in Computational Linguistic tasks. We wanted to ensure that using this measure would benefit our algorithm which is supported by the findings in section 4.2.2.3. The evaluation in section 4.2.3 supports the hypothesis H3b where the use of a statistical based summarisation measure can produce promising results in extracting the appropriate sentences to formulate an answer.

5.2.2 Topic signature generation

5.2.2.1 Originality

So far in the literature Topic signatures were developed using human intervention. Although this can increase the quality of the topic signature, there are cases where these linguistic entities are not utilised by mainstream applications due to the expense or lack of a domain expert to contribute to their development.

In our algorithm as described in 3.4.6.2 we identified an alternative statistical approach to picking topic terms, which is an alternative approach to what is being used so far, since the final topic selection depends on the statistical weight of the term and not expert intervention. Also using local techniques, we extended the current techniques in order to assign signature terms to the extracted topics.

The results as shown in 4.2.1.4 provided scores of 100% precision, 87.9% recall and an F-score of 0.935 for topic extraction. This is a very good score and in line with previous implementations where human intervention is part of the extraction algorithm. In the next step of our algorithm we used the topics extracted to populate them with signature terms. This so far is been done in a semi-automatic way where the expert was using statistical approaches to retrieve top signature terms and then develop the topic signatures from them.

In our work we looked into the distribution of frequencies of the potential signature terms and selected the words that fall outside the normal distribution as described in 3.4.6.2. This algorithm produced an 89.72% precision which is very competitive compared to the previous processes averaging 76% (Wang, 2004). The main advantage of this approach is that we can develop a knowledge base for a domain that we have a document collection for without using a domain expert with high accuracy. These results support our fourth hypothesis (H4) and can open up further research in the domain of automatically creating portable knowledge bases.

5.2.2.2 Hypothesis review

H4. Topic signatures can be acquired and used for computational tasks, using local analysis techniques and statistical weights without the intervention of an expert user.

Hypothesis 4 states that within the system a method to develop a dynamic knowledge base is required to operate without any human intervention. Topic signatures as described in 2.3.3 contain a topic term, which is an important term in a domain and signature terms which are terms related to the topic. The method described in 3.4.6 uses a two-step approach in order to develop topic signatures. The first is to identify topic terms and the second to assign signature terms to the topics.

The evaluation of the algorithm can be found in chapter 4.2.1.4, where using an empirical value of IDF >2, a precision of 100% is achieved for the task of identifying topic terms with recall of 87.9% which gives an F score of 0.9. Regarding the identification of topic terms in a learning object the scores achieved are high and provide confidence in the algorithm. Some further work may be needed to identify the empirical value automatically. In section 3.3.6.1 an explanation is given on why the specific value was picked and a generalisation of this method is feasible since the value of the IDF threshold depends on the corpus length.

The second part of the experiments that support Hypothesis 4 is to identify signature terms. This is a much more complicated task. This is because in a learning document, domain terms are closely connected to each other since learning materials describe specific theories, laws or properties where there is a strong link between the terms. The method employed to

identify the signature terms, used the distribution of the signature terms weights is described in Chapter 3.4.6. The specific approach is unique for the acquisition of topic signatures and for some questions the precision reached 100% with a minimum precision for one of the questions being 55%. The average precision of the algorithm was 89% which for such complicated task supports strongly the hypothesis.

The next section will explain about the originality of the project with respect to supporting students in a Virtual Learning Environment and also how hypothesis 5 is supported by the evaluation results.

5.2.3 Students receive correct information quicker and with less steps

5.2.3.1 Originality

There are approaches to provide Question Answering systems to be used with Virtual Learning Environments, as described in Chapter 2, but there is none to our current knowledge that accomplishes what the combination of algorithms described in Chapter 3 do. That is, without the use of a knowledge base or expert knowledge and purely using statistical weights and algorithms, the correct answer from a document can be returned to the user.

In current Virtual Learning Environments, student support by information retrieval is largely limited to search engines that weight documents and return the highest weighted document. Using the system described in this thesis, the student can ask questions to the system which will return a section of a document as the preferred answer.

From a user perspective, all the test users preferred the system when they were asked to choose between the question answering system and a search engine.

In the following section, the evidence that support hypothesis 5 is described.

5.2.3.2 Hypothesis Evaluation

H5. Using the Question Answering and Automatic summarisation techniques, students will be able to get the correct answer quicker and looking in fewer places than using standard search engines

The main goal of this work was the fifth hypothesis but in order to investigate and support it, a series of experiments needed to be conducted, in order to support staged hypotheses, to ensure the correct answer can be returned to the users. This is needed to ensure that there is a core system that can perform equally well or better than existing systems. As described above hypotheses 1 to 4 are strongly supported from the experiments described above.

Hypothesis 5 concentrates on the actual student support issue and for that reason it is evaluated with a user experiment as explained in chapter 4.3. Users, when they are given a list of documents, in order to find an answer may not be able to identify the correct one. If this is the only support mechanism available to them through a Virtual Learning Environment the student may retrieve and use the wrong information and also spend valuable time trying to retrieve information. When this is part of the learning task this may be appropriate, but in cases when the student requires to

retrieve correct information without looking at many documents, the support that search engines provide is limited.

In the user experiment described in chapter 4.3, the users were initially asked to answer the questions using the search engine. On average less than four questions out of the ten asked were correctly answered by the students. To achieve that low score, they also had to open up to 3 documents per question in some cases, sometimes do multiple searches and the average time spent per question ranged from 0.5 minute to 3 minutes.

The approach described in this thesis will provide the user with the correct answer if the correct sentences are selected by our modules in the minimum time possible (about 1 second to answer a question compared to looking into a list of document to retrieve the answer). Also the student does not have to navigate through the documents to look for the answer.

For the reasons stated above, it is clear that hypothesis 5 is supported by the evidence from the evaluation. The system is well preferred by the users.

In the next section, future enhancement to the system are described which will conclude the thesis.

5.3 Future work

One of the areas mentioned in the conclusion of the thesis that will need further investigation is to replace the stop word list with a more generic solution. An alternative approach could be to use the term weight in relation to the other terms of the corpus and filter out terms that have a low weight. Even a simple metric such as TF.IDF would be sufficient for such filtering.

On the other hand an extra calculation is required for this step which will introduce some delay to the system. A way to overcome the slower response time is to pre-process all the potential terms in the database and tag the ones with specific weights as stop words to show to the Query Parsing module that the word does not have any meaningful content.

The work done on the topic signatures is novel in a subject that has not be looked at very deeply in the Information Retrieval community. More experiments in different domains are necessary in order to confirm the hypotheses in other domains.

Finally, as we can see from the description of the final algorithm, there are a few parameters in that contribute in the retrieval of the correct answer. For example, apart from the statistical score, we use some local analysis on how important is the word in a subset of the corpus using IDF. Also if a topic signature is present in the question, then it is biased towards documents that contain the sequence. On the summarisation algorithm, we use the density and the length of the passage. These parameters can be measured using a machine learning approach and identify the importance of each metric in a formal way. This would make the combination of the parameters used more accurate for the users.

6 References

Agarwal, A., Raghavan H., Subbian, K., Melville, P., Lawrence, R., C. Gondek, D, Fan, J.(2012). Learning to rank for robust question answering. In Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12). ACM.

Agirre E., Ansa O., Hovy E., Martinez D.(2000). Enriching Very Large Ontologies using the WWW. In Proceedings of the First Workshop on Ontology Learning OL'2000 Berlin, Germany, August 25, 2000.

Amati G., Van Rijsbergen C. J.(2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Systems 20: pp. 357-389

Barrera A., Rakesh Verma R.(2011). Automated extractive single-document summarization: beating the baselines with a new approach. In Proceedings of the 2011 ACM Symposium on Applied Computing.

Berger A. et al.(2000). Bridging the Lexical Chasm: Statistical Approaches to Answer Finding. Proc. Int. Conf. Research and Development in Information Retrieval: pp. 192-199.

Biryukov M., Angheluta R., Moens M.(2005). Multidocument Question Answering Text Summarization Using Topic Signatures. Journal on Digital Information Management (JDIM) Volume 3.Issue 1

Bhowan U., McCloskey D.(2015). Genetic Programming for Feature Selection and Question-Answer Ranking in IBM Watson. 18th European Conference, EuroGP 2015, At Copenhagen, Denmark, Volume: 9025

Bobrow D.G.(1964). A question-answering system for high school algebra word problems. Pro. AFIPS Fall Joint Comput. Conf., Vol. 26, Pt. 1, Spartan Books, New York, pp. 591- 614

Brusilovsky P.(2004). KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. WWW 2004.

Buckland M., Gey F.(1994). The relationship between Recall and Precision. Journal of the American Society for Information Science. Volume 45,Issue 1, pages 12–19

Buitelaar P., Cimiano P., Magnini B.(ed.).(2005). Ontology Learning from Text: Methods, Evaluation and Applications. Volume 123 Frontiers in Artificial Intelligence and Applications

Brill E., Lin, J., Banko M., Dumais, S., Ng, A.(2001).Data-Intensive Question Answering. IN PROCEEDINGS OF THE TENTH TEXT RETRIEVAL CONFERENCE (TREC-10)

Brill E., Banko M., Dumais, S..(2002). An Analysis of the AskMSR Question-Answering System. In Proceedings of Empirical Methods in Natural Language Processing.

British Standard.(2003).BS8426: A code of practice for e-support in e-learning systems. ISBN 0 580 42450 2

Bowerman C., Stamoulos M., Oakes M.(2008) Generating topic signatures from a corpus of e-learning content. In: 6th International conference on Practical Applications in Language and Computers: PALC 2007, 19-22 Apr 2007, Lodz University Conference Centre, Poland.

Biryukov M, R. Angheluta R., Moens M. F.(2005). Multi-document question answering text summarization using topic signatures. Journal on Digital Information Management

CaoJ., Roussinov D., Robles-Flores J. A., Nunamaker J.(2005).

Automated question answering from lecture videos: NLP vs. pattern

matching. System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii

Ferrucci, D. et al.(2010).Building Watson: An Overview of the DeepQA Project.AI magazine. Vol 31.Issue 3.

Kumar C., Pingali P., Varma V.(2009) A light-weight summarizer based on language model with relative entropy. In Proceedings of the 2009 ACM symposium on Applied Computing (SAC '09). ACM, New York, NY, USA.

Lin C.Y., Hovy E.(2002). Automated multi-document summarization in NeATS. In Proceedings of the Human Language Technology Conference.

Carpineto C., Romano G.(2012). A Survey of Automatic Query Expansion in Information Retrieval. ACM Computer Surveys. 44, 1, Article 1 (January 2012)

Chali Y., Joty S.R., Hasan S.A.(2009).Complex Question Answering: Unsupervised Learning Approaches and Experiments. Journal of Artificial Intelligence Research. 35: pp.1-47.

Cuadros M., Padro L., Rigau G. (2005). Comparing methods for automatic acquisition of topic signatures. Recent Advances in Natural Language Processing (RANLP). Bulgaria. Borovets. 21-23 September. 2005,pp. 181-186.

Davis J., Goadrich M.(2006). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (ICML '06). ACM, New York, NY, USA, pp.233-240

Donghui F, Shaw E., Jihie K., Hovy E. (2006). An Intelligent Discussion-Bot for Answering Student Queries in Threaded Discussions .International Conference on Intelligent User Interfaces IUI-2006

Dunning T.(1993). Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19: pp.61-74.

Firth J. R. (1957) Papers in Linguistics. London. Oxford University Press.

Fisher S., Roark B.(2006). Query-focused summarization by supervised sentence ranking and skewed word distributions. In Proc. DUC-2006, New York, USA.

Erkan G, Radev D.(2004). Lexpagerank: Prestige in multidocument text summarization. In Proceedings of EMNLP

Gelbukh et al.(2010). Automatic Term Extraction Using Log-Likelihood Based Comparison with General Reference Corpus

Gonzalo J., Verdejo F., Chugur I., & Cigarran J.(1998). Indexing with WordNet synsets can improve text retrieval. Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP .Montreal, Canada, pp. 38-44.

He T., Zhang X., Xinghuo Y.(2006). An Approach to Automatically Constructing Domain Ontology. In: PACLIC 2006, Wuhan, China, November 1-3, pp. 150-157

Heie M., Whittaker E., Furui S.(2010). Optimizing Question Answering Accuracy by Maximizing Log-Likelihood. Proceedings of the ACL 2010 Conference Short Papers, pages 236–240,Uppsala, Sweden, 11-16 July 2010. Association for Computational Linguistics

Hildebrandt W., Katz B., Lin J.(2004). Answering definition questions using multiple knowledge sources. IN PROCEEDINGS OF HLT-NAACL 2004.

Hovy E., Lin C.(1996). Automated text summarization and the SUMMARIST system. Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, pp.197-214.

Kendel M.(2012). Internet global growth: lessons for the future. Analysys Mason.

Kumar P. et al(2005).A Fully Automatic Question-Answering System for Intelligent Search in E-Learning Documents. International Jl. on E-Learning.4(1).pp. 149-166

Kwok C., Etzioni, O., Weld D.(2001). Scaling question answering to the web. ACM Transactions on Information Systems, 19(3): pp. 242–262.

Moore, R.(2004). Improving IBM word-alignment model 1. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL '04). Association for Computational Linguistics, Stroudsburg, PA, USA, , Article 518

Lam-Adesina A. M., Jones G. J. F. (2001). Applying summarization techniques for term selection in relevance feedback. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New Orleans, Louisiana, USA, pp. 1-9.

Leacock C., et al.(1998).Using Corpus Statistics and WordNet Relations for Sense Identification. Computational Linguistics 24(1): pp.147-166.

Lin C.(2004).Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp.74-81

Lin C., Hovy E.(2000). The Automated Acquisition of Topic Signatures for Text Summarization. COLING.

Mihalcea R., Tarau P.(2004).Textrank: Bringing order into texts.In Proceedings of the Conference on Empirical Methods in Natural Language Processing.

Mihalcea R., Tarau P.(2005). An algorithm for language independent single and multiple document summarization. In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP).

Meena, Y. K., Gopalani D.(2014). Analysis of Sentence Scoring Methods for Extractive Automatic Text Summarization. In Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '14). ACM, New York, NY, USA.

Moschitti A.(2003). Answer filtering via text categorization in question answering systems. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence.pp. 241-248.

Nesi H. et al.(2007). British Academic Written English Corpus. University of Oxford Text Archive

Ouyang Y., Li S., Li W.(2007).Developing learning strategies for topic-based summarization. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM '07). ACM, New York, NY, USA, 79-86.

Ouyang Y., Li W., Zhang R., Li S., Lu Q.(2011). Applying regression models to query-focused multi-document summarization. Inf. Process. Manage. 47, 2.

Ouyang Y., Li W., Zhang R., Li S., Lu Q.(2013).A progressive sentence selection strategy for document summarization. Information Processing & Management, Volume 49, Issue 1, January 2013, Pages 213-221.

Prager J.,Chu-Carroll J., Czuba K.(2001). Use of WordNet Hypernyms for Answering What-Is Questions. Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, pp. 309–316.

Radev D. et al.(2004). MEAD - a platform for multidocument multilingual text summarization. In LREC, Lisbon, Portugal.

Ramos J. (2003).Using TF-IDF to Determine Word Relevance in Document Queries.The First instructional Conference on Machine Learning (iCML-2003)

Roussinov D., Fan W., Robles-Flores J.(2008). Beyond Keywords: Automated Question Answering on the Web. Communications of the ACM VOL. 51.NO. 9

Sahami M., Heilman T.(2006). A web-based kernel function for measuring the similarity of short text snippets. In Proceedings of the 15th international conference on World Wide Web (WWW '06).

Salton G. & Buckley C. (1988). Term-weighing approaches in automatic text retrieval. Information Processing & Management, 24(5): pp. 513-523.

Sedgewick R., Kevin W. (2008). Introduction to programming in Java : an interdisciplinary approach. Boston: Pearson Addison-Wesley, 2008. [ONLINE]. http://algs4.cs.princeton.edu/35applications/stopwords.txt

Soricut R., Brill E. (2006). Automatic question answering using the web: Beyond the factoid. In Journal of Information Retrieval-Special Issueon Web Information Retrieval, Vol. 9(2) pp. 191-206.

Sparck Jones, K. (1998). Information retrieval: how far will really simple methods take you? In: Proceedings TWTL 14, Twente University, the Netherlands, pp. 71-78.

Streiter O. et al. (2003). Example-based Term Extraction for Minority Languages: A case-study on Ladin. EURAC, Italy

Silva G., Ferreira R., Dueire Lins R., Cabral L., Oliveira H., Simske S., Riss M. (2015). Automatic Text Document Summarization Based on Machine Learning. In Proceedings of the 2015 ACM Symposium on Document Engineering (DocEng '15). ACM, New York, NY, USA.

Surdeanu M., Ciaramita M., Zaragoza H. (2011). Learning to rank answers to non-factoid questions from web collections. Comput. Linguist. 37, 2.

Suzuki J., Sasaki Y., Maeda E. (2002). SVM answer selection for open-domain question answering. In Proceedings of the 19th international conference on Computational linguistics - Volume 1 (COLING '02), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA.

Temperley D., Sleator D., Lafferty, J. (1993). Parsing English with a link grammar. Third Annual Workshop on Parsing Technologies

Unger C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A, Gerber, D., Cimiano, P. (2012). Template-based question answering over RDF data. In Proceedings of the 21st international conference on World Wide Web (WWW '12). ACM.

Wang Y., Wang W., Huang C. (2007). Enhanced Semantic Question Answering System for e-Learning Environment. 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)

Willliams, J. ,Goldberg, M.(2005). The evolution of eLearning. 22nd ascilite conference, Brisbane, Australia

Xia T. et al. (2013). E-Leaming Support System aided by VSM based Question Answering System. The 8th International Conference on Computer Science & Education (ICCSE 2013) April 26-28, 2013. Colombo, Sri Lanka

Xu J., Croft W. B. (1996). Query expansion using local and global document analysis. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '96). ACM, New York, NY, USA, pp. 4-11

Xu J., Licuanan A., Weischedel R. (2004). TREC 2003 QA at BBN: Answering Definitional Questions. Proceedings of TREC 2003.

Xu J., Croft W.,B. (2000). Improving the effectiveness of information retrieval with local context analysis. ACM Trans. Inf. Syst. 18, 1 (January

Wang X. (2004). Automatic acquisition of English topic signatures based on a second language, Proceedings of the ACL 2004 workshop on Student research, pp.49-es, July 21-26, 2004, Barcelona, Spain

Wang D., Zhu S., Li T., Gong Y. (2012). Comparative document summarization via discriminative sentence selection. ACM Trans. Knowl. Discov. Data 6, 3, Article 12.

Woods W. (1973). Progress in Natural Language Understanding: An Application to Lunar Geology. In Proceedings of the National Conference of the American Federation of Information Processing Societies, p.441-450.

Yao X., Van Durme B. (2014). Information Extraction over Structured Data: Question Answering with Freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 956–966

Zhang K., Zhao J. (2010). A Chinese question answering system with question classification and answer clustering. In Proceedings of IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Vol.6, 2010, pp. 2692-2696.

Appendixes

User evaluation raw data

User ID	User Clicks	User searches	Question	Start Time(ms)	End Time(ms)
student1	1	1	0	433283614	433493083
student2	2	3	0	433276645	433522194
student3	1	1	0	433275674	433351003
student4	3	1	0	433308679	433532835
student5	2	1	0	433547414	433646126
student6	1	1	0	433285310	433574174
student1	1	1	1	433503524	433599447
student2	1	1	1	433534351	433565665
student3	1	1	1	433369732	433414000
student4	1	1	1	433606052	433738282
student5	2	1	1	433710645	433845726
student6	4	1	1	433619962	433831253
student1	1	1	2	433604124	433665193
student2	1	1	2	433577678	433609425
student3	1	1	2	433437636	433484266
student4	6	1	2	433839695	434137711
student5	1	1	2	433856826	433974557
student6	2	1	2	433863272	434033331
student1	1	1	3	433668649	434009271
student2	3	1	3	433621668	433685152
student3	1	1	3	433488267	433545949
student4	1	1	3	434163301	434192130
student5	1	1	3	433983109	434009103
student6	1	1	3	434054598	434101696
student1	1	1	4	434013405	434029583
student2	1	2	4	433691448	433736895
student3	1	1	4	433549100	433582166
student4	3	1	4	434210308	434396851
student5	1	1	4	434018551	434051389
student6	1	1	4	434142743	434255624
student1	1	1	5	434031972	434058183
student2	2	1	5	433740731	433834654
student3	1	1	5	433622583	433641320
student4	1	1	5	434419950	434514581
student5	2	2	5	434063004	434260180
student6	1	1	5	434284390	434355836
student1	1	1	6	434062063	434081518
student2	1	1	6	433842479	433878518
student3	1	1	6	433647070	433669869

User ID	User Clicks	User searches	Question	Start Time(ms)	End Time(ms)
student4	2	1	6	434537675	434635025
student5	1	1	6	434261881	434398675
student6	1	1	6	434368019	434408916
student1	1	1	7	434084793	434118342
student2	1	1	7	433892320	433957965
student3	1	1	7	433743265	433767915
student4	1	1	7	434700084	434784522
student5	1	1	7	434414060	434481003
student6	1	1	7	434436300	434526528
student1	1	1	8	434120297	434146494
student2	1	1	8	433959663	433975605
student3	1	2	8	433781463	433807964
student4	1	1	8	434797419	434824119
student5	1	1	8	434496598	434528450
student6	1	1	8	434558359	434604458
student1	2	4	9	434148798	434207766
student2	1	1	9	433979422	434009405
student3	1	1	9	433826613	433847097
student4	1	1	9	434839235	434866002
student5	1	1	9	434537362	434630756
student6	1	1	9	434643657	434693257

User selected questions

User Id	Question	Selected Answer
student1	0	CHAPID=knet-1053022401471\RLOID=knet-1062612261218\RIOID=knet-1062612262000\knet\1053022401471\content.html.txt type=
student2	0	CHAPID=knet-1053022401471\RLOID=knet-1062612261218\RIOID=knet-1062612261812\knet\1053022401471\content.html.txt type=
student3	0	CHAPID=knet-1053022401471\RLOID=knet-1062612261218\RIOID=knet-1062612262000\knet\1053022401471\content.html.txt type=
student4	0	CHAPID=knet-1053022401471\RLOID=knet-1062612261218\RIOID=knet-1062612261812\knet\1053022401471\content.html.txt type=
student5	0	CHAPID=knet-1053022401471\RLOID=knet-1062612261218\RIOID=knet-1062612261812\knet\1053022401471\content.html.txt type=
student6	0	CHAPID=knet-1053022401471\RLOID=knet-1062612261218\RIOID=knet-1062612261812\knet\1053022401471\content.html.txt type=
student1	1	CHAPID=knet-1061921695747\RLOID=knet-1062626766906\RIOID=knet-1062626767484\knet\1061921695747\content.html.txt type=
student2	1	CHAPID=knet-1061921695747\RLOID=knet-1062626766906\RIOID=knet-1062626767484\knet\1061921695747\content.html.txt type=
student3	1	CHAPID=knet-1061921695747\RLOID=knet-1062626766906\RIOID=knet-1062626767484\knet\1061921695747\content.html.txt type=
student4	1	CHAPID=knet-1061921695747\RLOID=knet-1062626766906\RIOID=knet-1062626767484\knet\1061921695747\content.html.txt type=
student5	1	CHAPID=knet-1061921695747\RLOID=knet-1062626766906\RIOID=knet-1062626767484\knet\1061921695747\content.html.txt type=
student6	1	CHAPID=knet-1061921695747\RLOID=knet-1062626766906\RIOID=knet-

User Id	Question	Selected Answer
		1062626766968\knet\1061921695747\content.html.txt type=
student1	2	CHAPID=knet-1061921695747\RLOID=knet-1062626764921\RIOID=knet-1062626765843\knet\1061921695747\content.html.txt type=
student2	2	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761388046\knet\1061921696348\content.html.txt type=
student3	2	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761396000\knet\1061921696348\content.html.txt type=
student4	2	CHAPID=knet-1062175353187\RLOID=knet-1063148715265\RIOID=knet-rioov1063148715265\knet\1062175353187\content.html.txt type=
student5	2	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761388046\knet\1061921696348\content.html.txt type=
student6	2	CHAPID=knet-1061921696528\RLOID=knet-1077687089635\RIOID=knet-riosm1077687089635\knet\1061921696528\content.html.txt type=
student1	3	CHAPID=knet-1061921695747\RLOID=knet-1062626764750\RIOID=knet-rioov1062626764750\knet\1061921695747\content.html.txt type=
student2	3	CHAPID=knet-1061921696869\RLOID=knet-1061921696879\RIOID=knet-rioov1061921696879\knet\1061921696869\content.html.txt type=
student3	3	CHAPID=knet-1061921695747\RLOID=knet-1062626767921\RIOID=knet-1062626768218\knet\1061921695747\content.html.txt type=
student4	3	CHAPID=knet-1061921695747\RLOID=knet-1062626767921\RIOID=knet-1062626768218\knet\1061921695747\content.html.txt type=
student5	3	CHAPID=knet-1061921695747\RLOID=knet-1062626767921\RIOID=knet-1062626768218\knet\1061921695747\content.html.txt type=
student6	3	CHAPID=knet-1061921695747\RLOID=knet-1062626767921\RIOID=knet-

User Id	Question	Selected Answer
		1062626768218\knet\1061921695747\content.html.txt type=
student1	4	CHAPID=knet-1061921696148\RLOID=knet-1062807120093\RIOID=knet-1062807120765\knet\1061921696148\content.html.txt type=
student2	4	CHAPID=knet-1061921696148\RLOID=knet-1062807120093\RIOID=knet-1062807120593\knet\1061921696148\content.html.txt type=
student3	4	CHAPID=knet-1061921696148\RLOID=knet-1062807120093\RIOID=knet-1062807120765\knet\1061921696148\content.html.txt type=
student4	4	CHAPID=knet-1061921696148\RLOID=knet-1062807120093\RIOID=knet-1062807120453\knet\1061921696148\content.html.txt type=
student5	4	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-1062800990781\knet\1061921695948\content.html.txt type=
student6	4	CHAPID=knet-1061921696148\RLOID=knet-1062807120093\RIOID=knet-1062807120453\knet\1061921696148\content.html.txt type=
student1	5	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-1062800989046\knet\1061921695948\content.html.txt type=
student2	5	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-1062800990031\knet\1061921695948\content.html.txt type=
student3	5	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-1062800990031\knet\1061921695948\content.html.txt type=
student4	5	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-1062800989046\knet\1061921695948\content.html.txt type=
student5	5	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-1062800989046\knet\1061921695948\content.html.txt type=
student6	5	CHAPID=knet-1061921695948\RLOID=knet-1062800988968\RIOID=knet-

User Id	Question	Selected Answer
		1062800990031\knet\1061921695948\content.html.txt type=
student1	6	CHAPID=knet-1061921696528\RLOID=knet-1063922465062\RIOID=knet-1063922467140\knet\1061921696528\content.html.txt type=
student2	6	CHAPID=knet-1061921696709\RLOID=knet-1063149375359\RIOID=knet-1063149376078\knet\1061921696709\content.html.txt type=
student3	6	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761394093\knet\1061921696348\content.html.txt type=
student4	6	CHAPID=knet-1061921696709\RLOID=knet-1063149375359\RIOID=knet-1063149375625\knet\1061921696709\content.html.txt type=
student5	6	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761393609\knet\1061921696348\content.html.txt type=
student6	6	CHAPID=knet-1061921696709\RLOID=knet-1063149375359\RIOID=knet-1063149376078\knet\1061921696709\content.html.txt type=
student1	7	CHAPID=knet-1061921695948\RLOID=knet-1062800993609\RIOID=knet-1062800994468\knet\1061921695948\content.html.txt type=
student2	7	CHAPID=knet-1061921696528\RLOID=knet-1063922465062\RIOID=knet-1063922467140\knet\1061921696528\content.html.txt type=
student3	7	CHAPID=knet-1061921696528\RLOID=knet-1063922465062\RIOID=knet-1063922468031\knet\1061921696528\content.html.txt type=
student4	7	CHAPID=knet-1061921696709\RLOID=knet-1063149375359\RIOID=knet-1063149375875\knet\1061921696709\content.html.txt type=
student5	7	CHAPID=knet-1061921696528\RLOID=knet-1063922465062\RIOID=knet-1063922467437\knet\1061921696528\content.html.txt type=
student6	7	CHAPID=knet-1061921696709\RLOID=knet-1063149375359\RIOID=knet-

User Id	Question	Selected Answer
		1063149375421\knet\1061921696709\content.html.txt type=
student1	8	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761389953\knet\1061921696348\content.html.txt type=
student2	8	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761389953\knet\1061921696348\content.html.txt type=
student3	8	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761389953\knet\1061921696348\content.html.txt type=
student4	8	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761389953\knet\1061921696348\content.html.txt type=
student5	8	CHAPID=knet-1061921696348\RLOID=knet-1063761387937\RIOID=knet-1063761390406\knet\1061921696348\content.html.txt type=
student6	8	CHAPID=knet-1061921696348\RLOID=knet-1076708022202\RIOID=knet-riosm1076708022202\knet\1061921696348\content.html.txt type=
student1	9	CHAPID=knet-1061921696348\RLOID=knet-1063761402484\RIOID=knet-1063761406265\knet\1061921696348\content.html.txt type=
student2	9	CHAPID=knet-1061921695747\RLOID=knet-1062626764921\RIOID=knet-1062626765281\knet\1061921695747\content.html.txt type=
student3	9	CHAPID=knet-1061921697029\RLOID=knet-1062700912234\RIOID=knet-1062700912343\knet\1061921697029\content.html.txt type=
student4	9	CHAPID=knet-1061921697029\RLOID=knet-1062700912234\RIOID=knet-1062700912343\knet\1061921697029\content.html.txt type=
student5	9	CHAPID=knet-1061921697029\RLOID=knet-1062700912234\RIOID=knet-1062700912343\knet\1061921697029\content.html.txt type=
student6	9	CHAPID=knet-1061921697029\RLOID=knet-1062700912234\RIOID=knet-

User Id	Question	Selected Answer
		1062700912343\knet\1061921697029\content.html.txt type=

Results following bigram identification

Question	Bigram
which of the following describes the use of a network interface card (NIC)	network interface,interface card,NIC,following,Use,describes,
which of the following is used to describe the rated throughput capacity of a given network medium	given network,network medium,following,used,capacity,throughput,rated,
What describes a LAN	LAN,describes,
why was the OSI model created	OSI model,created,
why are the pairs of wires twisted together in UTP cable	UTP cable,wires,twisted,pairs,
What is required for electrons to flow?	required,flow,electrons,
How does using a hub or a repeater affects the size of collision domain	collision domain,does,using,size,repeater,Hub,affects,
Which of the following will cause a collision on an Ethernet network	Ethernet network,following,cause,collision,
which Ethernet implementations use rj-45 connectors	Ethernet implementations,Use,connectors,
which two functions of a router in a network	network,router,functions

Results following stopword removal.

which of the following describes the use of a network interface card (NIC)	network,interface,card,NIC,following,Use,describes,
which of the following is used to describe the rated throughput capacity of a given network medium	network,following,used,capacity,throughput,medium, given,rated,
What describes a LAN	LAN,describes,
why was the OSI model created	created,OSI,model,
why are the pairs of wires twisted together in UTP cable	cable,wires,UTP,twisted,pairs,

What is required for electrons to flow?	required,flow,electrons,
How does using a hub or a repeater affects the size of collision domain	does,using,size,repeater,Hub,Domain,affects,collision,
Which of the following will cause a collision on an Ethernet network	network,following,Ethernet,cause,collision,
which Ethernet implementations use rj-45 connectors	Use,Ethernet,implementations,connectors,
which two functions of a router in a network	network,router,functions,

Statistical results

		Q1			
Q1	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	814	99.042	16.507	261.396	43.566
	812	8.483	1.414	25.667	4.278
	970	14.814	2.469	58.768	9.795
	810	18.308	3.051	32.476	5.413
	813	58.703	9.784	123.501	20.583
	830	10.997	1.833	19.491	3.248
Q2	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	849	55.988	7.998	131.110	18.730
	845	17.331	2.476	48.976	6.997
	859	4.446	0.635	40.592	5.799
Q3	**Document ID**	**SumDC**	**AvgDC**	**SumSC**	**AvgSC**
	931	11.396	5.698	93.511	46.756
	919	13.496	6.748	72.283	36.141
	1010	9.571	4.786	67.943	33.972
	833	9.235	4.618	67.559	33.779
	907	8.528	4.264	66.739	33.369
	926	8.081	4.041	66.214	33.107
	982	6.835	3.417	64.714	32.357
	911	10.888	5.444	61.052	30.526
	991	8.296	4.148	53.963	26.982
	882	4.909	2.455	53.898	26.949
	964	6.933	3.466	52.423	26.212

	916	6.867	3.434	52.348	26.174
	920	5.586	2.793	50.840	25.420
	1053	4.609	2.304	49.637	24.818
	899	3.408	1.704	48.061	24.031
	858	3.002	1.501	38.417	19.209
	844	4.956	2.478	36.967	18.483
	910	4.513	2.256	36.459	18.229
	1016	1.416	0.708	35.892	17.946
	933	3.912	1.956	35.754	17.877
	912	3.038	1.519	34.681	17.341
	861	1.785	0.893	32.982	16.491
	905	1.480	0.740	32.518	16.259
	977	1.084	0.542	31.862	15.931
	934	1.019	0.509	31.748	15.874
	1054	0.865	0.432	31.465	15.732
	908	0.816	0.408	31.371	15.685
	890	5.867	2.934	31.368	15.684
	970	0.687	0.343	31.112	15.556
	870	0.217	0.109	29.928	14.964
	885	4.630	2.315	29.635	14.818
	848	3.772	1.886	25.344	12.672
	1041	2.362	1.181	23.476	11.738
	838	2.057	1.029	19.390	9.695
	840	2.030	1.015	19.358	9.679
	986	0.987	0.494	18.020	9.010
	1040	0.987	0.494	18.020	9.010
	839	0.921	0.460	17.924	8.962
	849	0.858	0.429	17.831	8.916
	966	0.800	0.400	17.743	8.872
	897	0.754	0.377	17.672	8.836
	832	0.670	0.335	17.538	8.769
	959	0.587	0.294	17.400	8.700
	811	0.546	0.273	17.329	8.664
	813	0.520	0.260	17.283	8.641
	904	0.520	0.260	17.283	8.641
	901	0.470	0.235	17.193	8.597
	850	0.458	0.229	17.171	8.586
	988	0.447	0.223	17.150	8.575
	883	0.441	0.221	17.139	8.570
	975	0.403	0.202	17.066	8.533
	962	0.353	0.177	16.966	8.483
	937	0.344	0.172	16.947	8.473
	1037	0.330	0.165	16.918	8.459
	877	0.295	0.148	16.843	8.422

	845	0.256	0.128	16.754	8.377
	961	0.241	0.121	16.719	8.359
	984	0.220	0.110	16.668	8.334
	922	0.210	0.105	16.643	8.321
	918	0.191	0.096	16.594	8.297
	881	0.104	0.052	16.336	8.168
	1011	0.066	0.033	16.198	8.099
	967	0.047	0.024	16.114	8.057
	880	0.016	0.008	15.937	7.969
	1051	0.011	0.006	15.902	7.951
	1001	0.006	0.003	15.851	7.926
	878	0.000	0.000	15.786	7.893
	835	-0.002	-0.001	15.743	7.871
	952	-0.005	-0.003	15.650	7.825
	992	-0.004	-0.002	15.595	7.798
	888	0.026	0.013	15.299	7.649
	859	0.030	0.015	15.282	7.641
	1046	5.670	2.835	14.651	7.326
	928	3.678	1.839	8.264	4.132
	914	3.352	1.676	7.917	3.958
	873	3.020	1.510	7.559	3.780
	996	2.911	1.456	7.440	3.720
	1017	2.744	1.372	7.256	3.628
	863	2.682	1.341	7.188	3.594
	853	2.636	1.318	7.137	3.569
	1033	2.494	1.247	6.979	3.490
	898	2.481	1.240	6.964	3.482
	995	2.364	1.182	6.833	3.416
	824	2.303	1.152	6.763	3.382
	956	2.155	1.078	6.594	3.297
	953	2.134	1.067	6.569	3.285
	924	2.123	1.061	6.557	3.278
	990	2.050	1.025	6.473	3.236
	942	1.907	0.953	6.305	3.152
	867	1.898	0.949	6.294	3.147
	886	1.862	0.931	6.252	3.126
	884	1.793	0.896	6.170	3.085
	1022	1.704	0.852	6.062	3.031
	856	1.657	0.829	6.006	3.003
	872	1.570	0.785	5.899	2.949
	1049	1.549	0.774	5.873	2.936
	972	1.468	0.734	5.772	2.886
	868	1.392	0.696	5.677	2.838
	1007	1.254	0.627	5.499	2.750

	1052	1.248	0.624	5.492	2.746
	894	1.227	0.613	5.464	2.732
	983	1.102	0.551	5.298	2.649
	864	1.065	0.532	5.248	2.624
	862	0.800	0.400	4.873	2.436
	1038	0.744	0.372	4.788	2.394
	876	0.555	0.277	4.489	2.244
	943	0.518	0.259	4.428	2.214
Q4	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	818	4.774	2.387	10.007	5.004
	832	5.383	2.692	-0.030	-0.015
	833	11.304	5.652	19.366	9.683
	834	1.575	0.788	4.099	2.050
	837	2.820	1.410	5.502	2.751
	839	2.657	1.329	5.325	2.662
	840	3.981	1.991	6.732	3.366
	845	1.658	0.829	4.197	2.098
	853	1.599	0.800	-0.008	-0.004
	854	37.002	18.501	11.066	5.533
	855	38.521	19.261	-0.459	-0.230
	856	4.120	2.060	-0.021	-0.011
	857	20.545	10.272	7.353	3.677
	858	6.876	3.438	-0.036	-0.018
	859	1.778	0.889	-0.010	-0.005
	863	14.744	7.372	22.920	11.460
	865	2.522	1.261	5.178	2.589
	866	0.832	0.416	3.153	1.577
	871	1.651	0.825	4.188	2.094
	875	1.716	0.858	4.265	2.132
	881	4.993	2.496	10.248	5.124
	888	0.617	0.308	2.843	1.421
	908	0.862	0.431	3.195	1.598
	912	2.147	1.073	4.760	2.380
	918	1.530	0.765	4.044	2.022
	919	1.625	0.813	-0.008	-0.004
	931	0.796	0.398	3.103	1.552
	933	6.122	3.061	-0.037	-0.019
	934	2.415	1.207	-0.012	-0.006
	935	1.127	0.563	-0.006	-0.003
	936	12.955	6.478	-0.081	-0.040
	937	0.999	0.500	-0.005	-0.003
	938	1.092	0.546	3.160	1.580
	940	1.589	0.794	4.115	2.057
	942	1.206	0.603	-0.006	-0.003

	952	0.266	0.133	-0.003	-0.001
	962	1.831	0.916	4.400	2.200
	963	0.261	0.130	-0.003	-0.001
	965	0.867	0.433	-0.005	-0.002
	967	0.532	0.266	-0.004	-0.002
	969	0.689	0.345	-0.004	-0.002
	970	2.917	1.458	3.065	1.533
	971	4.260	2.130	5.241	2.621
	975	1.206	0.603	-0.006	-0.003
	976	22.406	11.203	-0.195	-0.097
	977	2.245	1.123	-0.012	-0.006
	978	0.755	0.377	-0.004	-0.002
	979	5.904	2.952	11.237	5.619
	981	1.024	0.512	3.412	1.706
	984	1.589	0.794	4.115	2.057
	988	1.229	0.614	-0.006	-0.003
	996	2.060	1.030	-0.011	-0.005
	1001	0.335	0.167	-0.003	-0.002
	1004	1.976	0.988	4.566	2.283
	1010	1.848	0.924	4.419	2.209
	1011	4.751	2.375	9.981	4.991
	1012	1.651	0.825	4.188	2.094
	1016	0.264	0.132	-0.003	-0.001
	1018	1.257	0.629	-0.006	-0.003
	1027	2.871	1.435	-0.017	-0.008
	1029	1.467	0.733	-0.007	-0.004
	1030	1.197	0.598	3.635	1.817
	1033	1.652	0.826	-0.008	-0.004
	1036	0.612	0.306	-0.004	-0.002
Q5	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	869	35.298	8.824	85.828	21.457
	870	63.520	15.880	94.044	23.511
	888	6.384	1.596	30.063	7.516
	899	17.728	4.432	56.248	14.062
	901	44.465	11.116	109.335	27.334
	903	25.718	6.430	72.271	18.068
	905	19.828	4.957	61.010	15.252
	1041	19.380	4.845	18.904	4.726
	1054	7.137	1.784	25.386	6.347
Q6	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	857	0.036	0.012	3.385	1.128

	862	113.822	37.941	224.865	74.955
	863	32.605	10.868	69.026	23.009
	864	107.670	35.890	212.863	70.954
	865	30.509	10.170	66.832	22.277
	866	122.671	40.890	237.317	79.106
	888	10.603	3.534	41.622	13.874
	930	4.237	1.412	13.018	4.339
	963	0.384	0.128	4.727	1.576
	967	0.766	0.255	5.513	1.838
	1050	3.528	1.176	9.303	3.101
Q7	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	Row Labels	**SumDC**	**AvgDC**	**SumSC**	**AvgSC**
	857	0.036	0.012	3.385	1.128
	862	113.822	37.941	224.865	74.955
	863	32.605	10.868	69.026	23.009
	864	107.670	35.890	212.863	70.954
	865	30.509	10.170	66.832	22.277
	866	122.671	40.890	237.317	79.106
	888	10.603	3.534	41.622	13.874
	930	4.237	1.412	13.018	4.339
	963	0.384	0.128	4.727	1.576
	967	0.766	0.255	5.513	1.838
	1050	3.528	1.176	9.303	3.101
Q8	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	814	3.939	0.985	9.897	2.474
	861	2.097	0.524	5.952	1.488
	867	0.755	0.189	3.425	0.856
	869	2.392	0.598	6.352	1.588
	870	5.574	1.394	1.488	0.372
	875	5.015	1.254	12.256	3.064
	883	10.416	2.604	44.590	11.148
	886	30.466	7.616	34.359	8.590
	888	0.261	0.065	11.661	2.915
	890	2.527	0.632	6.530	1.633
	933	1.750	0.437	16.550	4.137
	934	0.113	0.028	2.248	0.562
	935	4.014	1.003	3.132	0.783
	938	2.182	0.545	2.054	0.513
	940	9.259	2.315	9.255	2.314
	941	8.340	2.085	41.023	10.256
	942	47.077	11.769	152.814	38.203
	943	22.474	5.618	120.870	30.217

	944	3.128	0.782	11.528	2.882
	946	84.600	21.150	303.279	75.820
	947	6.346	1.586	56.546	14.136
	948	5.355	1.339	27.587	6.897
	953	6.286	1.571	45.488	11.372
	954	14.076	3.519	98.030	24.508
	959	1.170	0.293	15.624	3.906
	961	56.068	14.017	208.433	52.108
	966	1.578	0.395	16.289	4.072
	985	1.451	0.363	4.991	1.248
	1005	3.434	0.859	10.326	2.581
	1038	6.817	1.704	56.751	14.188
Q9	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	810	2.785	0.928	9.669	3.223
	811	1.386	0.462	7.767	2.589
	813	0.051	0.017	2.569	0.856
	814	1.476	0.492	7.902	2.634
	815	0.171	0.057	2.931	0.977
	817	1.182	0.394	7.450	2.483
	818	8.819	2.940	29.465	9.822
	820	4.869	1.623	15.339	5.113
	821	1.037	0.346	4.377	1.459
	822	2.078	0.693	8.752	2.917
	823	0.359	0.120	3.331	1.110
	824	0.161	0.054	2.906	0.969
	827	0.960	0.320	7.086	2.362
	828	0.105	0.035	2.751	0.917
	830	0.016	0.005	2.406	0.802
	832	0.120	0.040	2.797	0.932
	833	0.878	0.293	6.944	2.315
	834	7.630	2.543	24.924	8.308
	835	0.312	0.104	1.294	0.431
	836	4.954	1.651	18.517	6.172
	841	0.474	0.158	3.535	1.178
	843	8.463	2.821	22.747	7.582
	844	0.552	0.184	3.667	1.222
	846	0.080	0.027	2.674	0.891
	848	0.359	0.120	3.331	1.110
	849	0.228	0.076	3.063	1.021
	850	0.028	0.009	2.471	0.824
	852	1.207	0.402	7.491	2.497
	854	-0.006	-0.002	2.226	0.742

	857	0.036	0.012	7.138	2.379
	861	1.988	0.663	11.087	3.696
	862	0.037	0.012	4.837	1.612
	866	0.387	0.129	1.217	0.406
	868	0.004	0.001	1.940	0.647
	869	0.695	0.232	6.611	2.204
	870	0.080	0.027	3.542	1.181
	876	4.390	1.463	26.436	8.812
	878	5.493	1.831	24.525	8.175
	880	10.737	3.579	38.672	12.891
	883	1.157	0.386	7.410	2.470
	884	1.061	0.354	7.254	2.418
	886	0.023	0.008	2.443	0.814
	888	0.324	0.108	8.269	2.756
	890	2.258	0.753	11.402	3.801
	893	1.936	0.645	11.420	3.807
	895	4.012	1.337	14.280	4.760
	899	5.630	1.877	23.668	7.889
	900	7.664	2.555	25.862	8.621
	901	2.597	0.866	11.788	3.929
	904	2.677	0.892	11.878	3.959
	906	7.028	2.343	25.187	8.396
	907	-0.007	-0.002	2.051	0.684
	908	1.339	0.446	10.283	3.428
	910	3.958	1.319	16.254	5.418
	911	0.296	0.099	3.207	1.069
	912	0.064	0.021	2.621	0.874
	913	19.215	6.405	3.834	1.278
	914	4.104	1.368	13.417	4.472
	915	8.378	2.793	39.173	13.058
	916	0.093	0.031	2.718	0.906
	917	0.019	0.006	2.424	0.808
	918	4.558	1.519	18.002	6.001
	920	0.939	0.313	7.050	2.350
	922	-0.004	-0.001	2.015	0.672
	923	0.007	0.002	1.921	0.640
	924	0.093	0.031	2.718	0.906
	925	11.842	3.947	29.839	9.946
	926	2.045	0.682	13.090	4.363
	927	0.116	0.039	2.786	0.929
	928	5.433	1.811	18.087	6.029
	929	7.728	2.576	27.395	9.132
	930	2.298	0.766	11.448	3.816
	931	9.108	3.036	22.796	7.599

	934	0.230	0.077	1.394	0.465
	935	-0.007	-0.002	2.210	0.737
	937	0.939	0.313	7.050	2.350
	940	-0.006	-0.002	2.036	0.679
	941	0.034	0.011	2.500	0.833
	943	0.743	0.248	0.949	0.316
	950	11.787	3.929	-0.011	-0.004
	952	0.379	0.126	1.224	0.408
	954	0.166	0.055	1.489	0.496
	956	1.588	0.529	8.066	2.689
	959	0.080	0.027	2.674	0.891
	960	0.253	0.084	3.119	1.040
	961	-0.008	-0.003	2.080	0.693
	962	0.960	0.320	7.086	2.362
	963	0.364	0.121	1.239	0.413
	967	0.114	0.038	1.581	0.527
	969	4.488	1.496	17.910	5.970
	970	5.892	1.964	25.684	8.561
	973	4.694	1.565	15.127	5.042
	976	0.108	0.036	2.763	0.921
	980	-0.004	-0.001	2.250	0.750
	981	1.076	0.359	9.968	3.323
	982	0.090	0.030	1.633	0.544
	983	0.250	0.083	5.616	1.872
	984	0.655	0.218	6.532	2.177
	986	2.329	0.776	9.085	3.028
	990	0.070	0.023	2.642	0.881
	994	4.337	1.446	14.688	4.896
	995	0.187	0.062	2.970	0.990
	997	-0.006	-0.002	2.226	0.742
	998	0.080	0.027	2.674	0.891
	1000	2.243	0.748	11.884	3.961
	1001	0.082	0.027	5.056	1.685
	1002	1.476	0.492	7.902	2.634
	1003	0.026	0.009	1.826	0.609
	1004	1.157	0.386	7.410	2.470
	1005	1.418	0.473	10.580	3.527
Q10	**Document ID**	**SumDC**	**AvgDC**	**Sum SC**	**Avg SC**
	817	0.233	0.0777	55.281	18.427
	828	2.4801	0.8267	45.253	15.084
	832	2.1507	0.7169	33.925	11.308
	835	19.194	6.398	273.52	91.174
	839	2.5381	0.846	24.138	8.0459
	842	11.701	3.9003	83.709	27.903

	855	14.105	4.7016	113.45	37.815
	856	3.5831	1.1944	77.576	25.859
	857	-0.029	-0.0096	104.16	34.72
	858	3.9463	1.3154	97.929	32.643
	859	23.029	7.6763	292.21	97.404
	861	0.6081	0.2027	41.407	13.802
	870	2.0281	0.676	63.693	21.231
	878	1.9607	0.6536	52.614	17.538
	888	4.7774	1.5925	36.666	12.222
	910	1.3745	0.4582	26.109	8.7029
	915	8.471	2.8237	77.857	25.952
	921	4.3391	1.4464	87.413	29.138
	922	10.491	3.4971	120.27	40.089
	923	15.812	5.2707	175.66	58.555
	926	27.077	9.0257	138.65	46.217
	927	20.903	6.9676	112.33	37.444
	929	27.114	9.0381	131.37	43.79
	931	12.442	4.1475	225.88	75.294
	938	5.8575	1.9525	17.42	5.8068
	940	5.1175	1.7058	11.367	3.789
	942	2.547	0.849	55.509	18.503
	955	1.188	0.396	48.267	16.089
	962	6.213	2.071	62.182	20.727
	964	4.985	1.6617	57.751	19.25
	965	0.0851	0.0284	64.37	21.457
	969	5.4981	1.8327	70.886	23.629
	971	8.3882	2.7961	55.948	18.649
	972	1.5068	0.5023	68.929	22.976
	974	6.9127	2.3042	53.211	17.737
	977	21.36	7.1201	239.89	79.963
	980	14.646	4.8821	177.98	59.326
	982	46.203	15.401	355.42	118.47
	983	9.838	3.2793	188.4	62.798
	986	1.8063	0.6021	59.572	19.857
	988	0.775	0.2583	66.899	22.3
	989	0.6432	0.2144	34.2	11.4
	992	19.528	6.5095	257.26	85.753
	995	7.8614	2.6205	109.11	36.37
	996	3.4529	1.151	37.122	12.374
	997	14.587	4.8622	91.002	30.334
	1000	28.635	9.5449	192.46	64.153
	1001	30.775	10.258	172.28	57.426
	1002	10.233	3.4109	100.01	33.335
	1003	22.1	7.3665	166.48	55.492

	1004	12.805	4.2682	80.848	26.949
	1005	18.463	6.1545	215.51	71.835
	1007	3.2495	1.0832	106.25	35.417
	1008	3.4631	1.1544	96.32	32.107
	1010	9.4787	3.1596	141.01	47.003
	1011	1.731	0.577	111.55	37.185
	1014	3.5476	1.1825	54.518	18.173
	1016	6.8514	2.2838	194.63	64.877
	1017	8.1102	2.7034	34.051	11.35
	1018	2.4484	0.8161	22	7.3335

Topic Signatures

Question 1

Signatures for topic card			**Signatures for topic NIC**		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
card	110.5704	1	nic	235.9758	1
nic	96.19742	0.870011	card	44.68026	0.189342522
pc	60.38121	0.546088	adapter	35.2971	0.149579324
board	44.01938	0.398112	ping	34.72083	0.147137241
internet	36.09418	0.326436	collisions	34.3755	0.145673826
connect	32.99679	0.298423	collision	34.15723	0.144748847
modem	29.46902	0.266518	pc	32.48583	0.137665941
serial	28.34884	0.256387	fluke	30.73628	0.130251837
motherboard	28.0039	0.253268	category	30.04222	0.127310568
interface	27.23872	0.246347	hubs	27.3357	0.11584112
dte	25.54555	0.231034	connector	25.24919	0.106999072
pcmcia	25.52319	0.230832	microprocessor	22.65044	0.095986275
ir	25.52319	0.230832	620	22.65044	0.095986275
nics	24.63249	0.222776	aui	22.65044	0.095986275
multicast	23.24613	0.210238	jam	22.42493	0.095030603
devices	22.77167	0.205947	motherboard	20.04808	0.084958182
connection	22.72668	0.20554	board	19.95783	0.084575723
adapter	21.29807	0.19262	pcmcia	19.41496	0.082275194
dce	20.74796	0.187645	bad	19.41496	0.082275194
printed	20.03468	0.181194	10base2	17.95227	0.076076754
microprocessor	20.03468	0.181194	nics	17.67768	0.07491311
modems	19.76531	0.178758	hub	17.37131	0.073614771
expansion	17.03208	0.154038	map	16.76026	0.071025322

Question 2

Signatures for topic network			Signatures for topic medium		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
network	476.8863	1	medium	176.8268	1
ip	87.77289	0.184054114	ethernet	118.6348	0.670909445
routing	82.8655	0.173763619	gigabit	67.57812	0.382171141
addresses	77.88541	0.163320699	mbps	58.95982	0.333432507
address	75.96366	0.159290903	10gbe	41.82234	0.236515807
subnet	59.9097	0.125626779	ieee	40.21721	0.22743835
devices	57.00954	0.119545334	10base5	34.38966	0.19448212
internet	54.05222	0.113344033	802	31.11915	0.175986594
broadcast	50.80804	0.106541201	topology	26.61387	0.150508099
class	43.08614	0.090348866	encoding	24.90223	0.140828343
access	34.10444	0.071514807	10	24.26581	0.13722923
protocol	30.37056	0.06368512	sqe	22.52179	0.127366364
mask	30.06681	0.063048177	forms	22.52179	0.127366364
arp	29.49072	0.061840138	legacy	22.17756	0.125419623
networks	26.04947	0.054624058	100	21.63109	0.122329216
protocols	25.33776	0.053131647	timing	20.40228	0.115379996
subnetting	24.87896	0.052169574	frames	18.73891	0.105973241
layers	23.84849	0.050008741	standard	17.94349	0.101474936
routers	22.17305	0.046495468	frame	17.50041	0.098969196

Signatures for topic throughput		
Term With Overuse	**Weight**	**LL Over Max LL**
bandwidth	122.3263565	1.158396797
throughput	105.5997019	1
802	31.77778483	0.300926842
window	31.57023742	0.298961426
11b	31.32030556	0.29659464
jam	28.46637393	0.269568696
acknowledgment	28.31778712	0.26816162
1000baset	22.87779816	0.216646427
dsss	22.02373489	0.208558684
wireless	21.56828374	0.204245688
size	20.28280051	0.192072517
windowing	18.97940077	0.179729681
join	18.36980656	0.173956993

scanning	18.36980656	0.173956993

Question 3

Signatures for topic LAN			Signatures for topic describes		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
lan	237.4751	1	describes	139.8088592	1
devices	62.85961	0.264699804	electrons	51.74473244	0.37011054
arp	49.74989	0.20949521	atoms	42.57153896	0.30449815
wireless	46.77616	0.196972929	ap	34.28015628	0.245193019
noise	31.39246	0.132192639	protons	33.45061379	0.239259615
transmitter	28.53545	0.120161848	linkstate	33.2079512	0.237523941
signals	28.28434	0.119104434	dhcp	33.2079512	0.237523941
switches	27.45466	0.115610698	antenna	32.8184047	0.234737662
802	27.18674	0.114482511	p	30.41011781	0.217512095
area	26.62586	0.11212065	nucleus	30.41011781	0.217512095
lans	26.52764	0.111707034	ref	24.32884447	0.174015042
wire	23.49272	0.098927076	helium	24.32884447	0.174015042
bandwidth	22.70494	0.095609793	atom	24.32884447	0.174015042

Question 4

Signatures for topic OSI			Signatures for topic model		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
osi	303.8774	1.000	model	394.8082	1
model	287.8596	0.947	osi	240.5157	0.609196437
layer	210.1806	0.692	layers	181.6259	0.460035914
layers	190.7889	0.628	layer	168.0572	0.425668094
tcp	87.22657	0.287	tcp	116.5169	0.295122953
models	68.55037	0.226	models	71.43883	0.180945689
application	45.80521	0.151	ip	59.23983	0.150047118
ip	43.45062	0.143	transport	53.14635	0.134613096
transport	42.98333	0.141	application	53.05466	0.134380866

mac	38.74018	0.127	reference	28.54025	0.072288901
structured	38.37926	0.126	rarp	24.63964	0.062409134

Signatures for topic created		
Term With Overuse	**Weight**	**LL Over Max LL**
created	136.4910859	1
subnet	62.07261165	0.454774107
current	43.59612911	0.31940642
browser	35.7898492	0.262213821
id	32.6143263	0.238948398
voltage	30.70718964	0.224975788
class	26.01484413	0.190597386
subnetting	25.82037292	0.189172595
field	25.75215999	0.188672834
dod	25.40705348	0.186144416
model	22.56479797	0.165320671

Question 5

Signatures for topic wires			**Signatures for topic twisted**			**Signatures for topic pairs**		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
wires	143.08	1.00	cable	268.48	2.74	cable	217.05	1.51
cable	109.00	0.76	twisted	98.14	1.00	pairs	143.96	1.00
wire	105.51	0.74	wire	73.01	0.74	wire	142.04	0.99
utp	66.99	0.47	stp	71.88	0.73	pair	138.73	0.96
pair	50.89	0.36	pair	67.97	0.69	crosstalk	108.05	0.75
category	49.34	0.34	utp	58.48	0.60	utp	85.07	0.59
structured	47.12	0.33	sctp	53.54	0.55	duplex	53.57	0.37
crosstalk	37.62	0.26	noise	46.59	0.47	noise	52.69	0.37
5e	37.08	0.26	crosstalk	38.68	0.39	category	43.52	0.30
pins	35.56	0.25	shield	38.37	0.39	core	43.36	0.30
pairs	34.24	0.24	shielded	29.64	0.30	stp	41.39	0.29
stp	30.19	0.21	coaxial	29.32	0.30	test	39.53	0.27

Question 6

Signatures for topic required			Signatures for topic flow			Signatures for topic electrons		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
required	168.08	1.00	flow	289.23	1.00	electrons	272.74	1.00
delay	46.30	0.28	electrons	173.01	0.60	resistance	124.88	0.46
routing	35.73	0.21	resistance	78.64	0.27	current	114.93	0.42
arp	32.04	0.19	current	67.96	0.23	charges	69.48	0.25
dte	28.85	0.17	transport	61.82	0.21	atoms	69.48	0.25
console	26.09	0.16	domains	44.55	0.15	voltage	65.03	0.24
link	25.53	0.15	bandwidth	44.49	0.15	flow	64.86	0.24
autonegotiation	25.17	0.15	atoms	44.07	0.15	protons	54.59	0.20
linkstate	23.26	0.14	charges	44.07	0.15	force	52.34	0.19
dhcp	23.26	0.14	broadcasts	43.02	0.15	nucleus	49.63	0.18
serial	21.95	0.13	layer	36.95	0.13	materials	43.22	0.16
dce	21.64	0.13	voltage	35.29	0.12	circuits	40.90	0.15
spacing	20.90	0.12	protons	34.63	0.12	atom	39.71	0.15
modem	20.55	0.12	control	33.24	0.11	helium	39.71	0.15

Question 7

Signatures for topic collision			Signatures for topic domain			Signatures for topic repeater		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
collision	540.13	1.00	collision	273.36	1.00	repeater	124.61	1.00
collisions	174.28	0.32	domain	234.82	0.86	collision	43.69	0.35
domains	165.14	0.31	domains	95.46	0.35	hubs	42.41	0.34
frame	139.90	0.26	broadcast	71.97	0.26	repeaters	41.81	0.34
domain	95.43	0.18	layer	68.79	0.25	ethernet	34.94	0.28
station	71.50	0.13	bridge	56.32	0.21	station	34.71	0.28
bridge	65.45	0.12	collisions	51.80	0.19	collisions	33.27	0.27
broadcast	63.36	0.12	frame	50.75	0.19	spacing	32.48	0.26
stations	55.91	0.10	table	47.85	0.18	rule	31.77	0.25
broadcasts	44.22	0.08	station	47.55	0.17	timing	29.31	0.24
error	39.69	0.07	delay	36.49	0.13	mbps	27.72	0.22
frames	39.33	0.07	2	33.70	0.12	bittimes	26.71	0.21
jam	38.45	0.07	broadcasts	29.88	0.11	hub	26.39	0.21
legal	38.45	0.07	10base5	29.27	0.11	collided	22.26	0.18

Signatures for topic hub			Signatures for topic affects		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
hub	133.46	1.00	narrowband	52.95	1.00
hubs	60.32	0.45	affects	52.95	1.00
console	54.80	0.41	noise	42.72	0.81
10baset	42.20	0.32	broadcast	41.16	0.78
photozoom	39.11	0.29	broadcasts	40.00	0.76
rj	37.95	0.28	jam	34.60	0.65
45	36.44	0.27	white	26.18	0.49
passive	29.20	0.22	multicast	25.77	0.49
devices	25.28	0.19	interference	25.00	0.47
architecture	24.94	0.19	radiation	21.22	0.40
jack	23.94	0.18	organized	21.22	0.40
connect	22.77	0.17	block	21.18	0.40
straightthrough	22.14	0.17	baseband	16.33	0.31
workstations	22.14	0.17	storms	16.32	0.31

Question 8

Signatures for topic Ethernet			Signatures for topic network			Signatures for topic collision		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
ethernet	754.5221	1	network	476.8863	1	collision	540.1277	1
collision	152.19	0.20	ip	87.77	0.18	collisions	174.28	0.32
gigabit	143.67	0.19	routing	82.87	0.17	domains	165.14	0.31
frame	143.31	0.19	addresses	77.89	0.16	frame	139.90	0.26
mbps	123.74	0.16	address	75.96	0.16	domain	95.43	0.18
station	78.77	0.10	subnet	59.91	0.13	station	71.50	0.13
10baset	75.19	0.10	devices	57.01	0.12	bridge	65.45	0.12
100	74.47	0.10	internet	54.05	0.11	broadcast	63.36	0.12
duplex	71.92	0.10	broadcast	50.81	0.11	stations	55.91	0.10
collisions	65.52	0.09	class	43.09	0.09	broadcasts	44.22	0.08
timing	62.43	0.08	access	34.10	0.07	error	39.69	0.07
cable	61.69	0.08	protocol	30.37	0.06	frames	39.33	0.07
category	56.89	0.08	mask	30.07	0.06	jam	38.45	0.07
delay	56.36	0.07	arp	29.49	0.06	legal	38.45	0.07
fcs	45.79	0.06	networks	26.05	0.05	spacing	37.96	0.07

Question 9

Signatures for topic Ethernet			Signatures for topic implementations			Signatures for topic connectors		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
ethernet	754.52	1.00	implementations	115.38	1.00	cable	174.57	1.00
collision	152.19	0.20	ethernet	108.40	0.94	fiber	130.68	0.75
gigabit	143.67	0.19	duplex	55.09	0.48	connectors	117.63	0.67
frame	143.31	0.19	gigabit	52.43	0.45	connector	74.73	0.43
mbps	123.74	0.16	synchronous	29.46	0.26	rj	59.23	0.34
station	78.77	0.10	half	28.49	0.25	crosstalk	55.74	0.32
10baset	75.19	0.10	station	28.22	0.24	impedance	49.04	0.28
100	74.47	0.10	timing	25.73	0.22	console	48.58	0.28
duplex	71.92	0.10	10gbe	24.82	0.22	45	46.49	0.27
collisions	65.52	0.09	vendors	22.20	0.19	fiberoptic	43.93	0.25
timing	62.43	0.08	3ae	22.20	0.19	receiver	38.71	0.22
cable	61.69	0.08	companies	21.15	0.18	optical	38.43	0.22
category	56.89	0.08	km	18.10	0.16	light	37.29	0.21

Question 10

Signatures for topic network			Signatures for topic router		
Term With Overuse	**Weight**	**LL Over Max LL**	**Term With Overuse**	**Weight**	**LL Over Max LL**
network	476.89	1.00	router	395.09	1.00
ip	87.77	0.18	routing	301.66	0.76
routing	82.87	0.17	routers	110.30	0.28
addresses	77.89	0.16	address	56.14	0.14
address	75.96	0.16	console	51.44	0.13
subnet	59.91	0.13	straightthrough	48.87	0.12
devices	57.01	0.12	crossover	48.87	0.12
internet	54.05	0.11	linkstate	44.04	0.11
broadcast	50.81	0.11	metrics	41.57	0.11
class	43.09	0.09	arp	40.42	0.10
access	34.10	0.07	hop	38.58	0.10
protocol	30.37	0.06	route	37.69	0.10
mask	30.07	0.06	packet	37.15	0.09
arp	29.49	0.06	rollover	36.01	0.09
networks	26.05	0.05	routed	34.62	0.09