Matching Text Meaning using Dependency Analysis and Word Group Similarities

by

Naomi Frankel

Third Year Project
Computer Science & Artificial Intelligence
Department of Informatics
University of Sussex
2007

Candidate Number: 83399

Under Supervision of Bill Keller

Statement of Originality

This report is submitted as part requirement for the degree of Computer Science and Artificial Intelligence at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged.

Naomi Frankel

Summary

The project attempts to develop and evaluate a method for matching pairs of texts with similar or equivalent meaning using statistical Natural Language Processing (NLP) techniques. Taking advantage of the high redundancy of information available today in repositories such as the World Wide Web, the method uses the grammatical relations between words, along with lexical similarity sets, to compare between a pair of sentences. The hypothesis is that this method should return more accurate results than a classic keyword-based search due to its use of grammatical relations, and that the use of related word sets will enable it to find the similarity between varying ways of phrasing the same meaning.

To evaluate the validity of the proposed method, a question answering application was developed and tested. This application uses an external parsing system to acquire the grammatical relations making up the sentence, and compares them using a set overlap measure of a set of related words. Subjective, objective and comparative tests were run on the application for the purpose of evaluating its output and drawing conclusion about the validity of the proposed method, its advantages and limitations. The tests revealed the reliance on proper grammar to be a weakness but pointed to a comparably high precision value as the method's strength.

Acknowledgments

I would like to thank John Carroll and Ted Briscoe for the use of the RASP parser which plays an integral part of the implemented system. I would also like to thank my helpful participants who contributed an important part to the evaluation of my system.
But above all I would like to thank my supervisor Bill Keller who guided me throughout the project with stimulating discussions, helpful feedback and a great deal of patience.

Table of content

1. Introduction

1.1 Overview of the project

The primary aim of the project is to develop and evaluate a method for matching pairs of texts with similar or equivalent meaning taking advantage of the high redundancy of information available today in repositories such as the web. The main aim of this paper is to evaluate the feasibility of the proposed method and the types of applications it could be applied for.
The proposed method uses a comparison between sets of related words on sentences with a similar grammatical structure. As its title suggests, it is based on matching sentences using dependency analysis and word set similarities. This is achieved by parsing each sentence in a text to find its grammatical structure and then comparing the words of sentences with a similar structure using statistical analysis of related word sets.

The hypothesis is that this method should work best on a large mass of data such as the World Wide Web due to the redundancy of information it contains, and that it would perform better than a word search method due to its focus on grammatical relations.

The method was evaluated by implementing a question answering application which uses the proposed text similarity measure to search for the best matching sentence from a database of texts. This application was then run on different types of input and its output evaluated in terms of absolute and comparative performance.

1.2 Motivation

Since the invention of the digital computer in the first half of the 20th century the human user has had to learn how to communicate with the computer and adjust the way he worked accordingly.
At first computing power was so small that it limited the types of input which could be processed making it impossible to include all parameters needed to use natural language. However, computers have since evolved to have substantially larger processing powers making the notion of communicating using natural language seemingly within our grasp.

Communication at any level is about the exchange of information. In order for a dialog to be productive the information must not only be passed from one side to the other, but some kind of processing must be done on it to provide a pertinent response.
Computers, as we do, have to have some way of comparing what they get as input to their accumulated knowledge (in the computer's case accumulated by its programmer rather than by its life experience). This project attempts to do so by proposing a method for matching pairs of texts with similar or equivalent meaning.

If successful, this method could have far reaching effects as there are so many different applications which would gain from it, from quite obvious ones like data mining applications to systems which require speech recognition. For example, a system which uses speech recognition to perform daily tasks in the home could accept an input such as "Make the room brighter" and recognize that by using the word 'brighter' the user is referring to the light source in the room (as bright and light are related words) and therefore find the relevant action in its database of commands.
Another possible application would be to use the method to compare between texts of different languages. Since the method uses the relation between words rather than the word's place in the sentence the relationship between 'blue' and 'house' in the English sentence 'Tom entered the blue house' would match the one between 'azul' and 'casa' in the Portuguese sentence 'Tom entrou na casa azul' despite the fact that their occurrence in the sentence is different.

1.3 Report Structure

This report will begin by considering the professional issues which may arise in a project of this type. I will then provide some background information which will assist in understanding this work and put it in context. This will include a short scan of the history of the field of Natural Language Processing (NLP) followed by introductions to the terms and concepts used in this report. To put the work in context I will then discuss some existing systems being used to solve similar problems, mentioning what they are lacking and how my solution will be different.

The next part of the report will discuss the requirements I have set for this system, both in terms of the proof of concept and in terms of what would make it useful to an end user of a system using this method (in this case the system implemented to test and demonstrate it).

The design section will begin by providing an overview of the processes the system uses before going into detail about each stage of the method.
Next I will describe the implementation of the system built to demonstrate the method discussing the technologies and data structures used.

Due to the nature of the project a key phase of the work has been the evaluation of the system output. Therefore the next section will discuss the different evaluation methods employed, and lay out the results of the evaluation experiments performed. The evaluation section will conclude with an analysis which will discuss what these results tell us about the validity of the method and what seem to be its advantages and limitations.

This will be followed by the known technological limitations of the implemented system and possible future work that can be done to take better advantage of the method strengths.
The paper will conclude with an assessment of the success of the project in terms of the experiment, the finished product and my personal gain.

2. Professional Considerations

Every work being done for the public domain is likely to have some kind of impact on its audience and therefore may raise ethical issues. To make sure all professionals take these issues into account ethical standards are set by a governing body. For the computing profession in Britain these standards are defined by the British Computer Society in the Code of Conduct (BCS 2006a) and Code of Good Practice (BCS 2006b).

Due to the research nature of this project, it will not have an immediate impact on a client base and therefore many of the issues raised in the two documents are not relevant. However, there are some issues which apply to nearly all work types and some that are relevant because of the project being a form of research.
The first issue and one which applies to most anyone undertaking a project is covered in section 15 of the 'Code of Conduct':

"*15. You shall not claim any level of competence that you do not possess. You shall only offer to do work or provide a service that is within your professional competence.*

You can self-assess your professional competence for undertaking a particular job or role by asking, for example,

1. *am I familiar with the technology involved, or have I worked with similar technology before?*
2. *have I successfully completed similar assignments or roles in the past?*
3. *can I demonstrate adequate knowledge of the specific business application and requirements successfully to undertake the work?*" (BCS 2006a: pp. 4)

Although my project is not undertaken with an end user in mind, the issue of professional competence and integrity is still one to be taken seriously as it is bound to have a great effect on its outcome. Because it is a research based project, the results of the study can hardly be taken seriously if the level of work is not appropriate.

The Code of Practice has slightly more bearing on this project as it goes into more detail about IT practices in general and research based work in particular. The first section relevant to this project is 3.1, Programme/Project Management:

"When Planning

Seek out similar projects and benefit from the lessons learned.

When Tracking Progress

Do not assume that any overruns can be recovered later in the project; in particular do not cut back on later activities such as testing." (BCS 2006b: pp. 11-12)

One of the first tasks I did when planning the project was a survey of the existing solutions to the problem I proposed to tackle. This is not only to provide myself and the reader of this report a background and context to the evaluation of this work but also to direct my thinking about the solution to the problem.

This entire quote is actually good advice rather than a standard a professional has to follow. The second part of the quote above is an invaluable lesson which one re-learns on every project and is likely to forget before the next. Good planning is the key to a successful project and a crucial point of the planning process is to make sure to set aside adequate time to all parts of the project. As this project does not involve a delivery to a client, it is easy to underestimate the importance of testing. But as in section 15 of the Code of Conduct, without it, the results of the project can hardly be taken seriously.

The second segment relevant to this project is part of section 4, Practices Specific to Education and Research Functions. Sub-section 4.2 discusses issues specifically relevant to research projects:

"When Performing Research

- *Recognise the potential use or misuse of the outcomes of your research and only proceed with the research if you can justify to yourself the consequences.*
- *Investigate the analysis and research by other people and organisations into related topics and acknowledge their contribution to your research."* (BCS 2006b: pp. 22)

The first point raises an issue which must be considered from the onset and which should be re-evaluated throughout a research project. It is also potentially a great problem as even the most seemingly innocent work may be harmful if used in a different context. The field of NLP as a whole can be seen in this way. Obviously it can have far reaching positive effects on our everyday use of computers, but it is also easy to imagine NLP applications being used to promote potentially harmful fields such as weapon manufacturing.

The second point taken from section 4.2 repeats the idea presented in section 3.1 of researching other work in the domain the project covers while reminding us of the importance of acknowledging any of the work being used or referred to. This point is highly important in my project as I make use of several existing resources as part of my implementation.

3. Background

3.1 History

Even before the beginning of the 20th century there was already work in progress, theoretical and practical, leading to the development of the modern computer.
However the first steps in what shaped the field of NLP can be recognised in the decoding efforts of the Second World War. The work done on the Enigma and other code breaking machines made people feel that using natural language in machines can be viewed as just another form of code cracking. This belief led the way to a wave of work in Machine Translation using word for word techniques. But there was little success in this field, leading researchers to realise how hard this problem actually is and to a period of very low funding for work in NLP.

In 1950, Alan Turing published his paper *Computing Machinery and Intelligence* (Turing, 1950) in which he introduced the idea of a test which, if passed, a machine can be said to have human behaviour. The test required the machine to try and fool a human user into thinking that they are chatting with another human rather than a machine, and therefore required the use of natural language. Turing suggested this test as an empirical way to evaluate whether a machine is thinking as a solution to the problem of defining what it means for a machine to think.

In 1957, Noam Chomsky, an American Linguist, published *Syntactic Structures* (Chomsky, 1957) which has become one of the most important publications in Linguistics and also had great effect on the field of NLP. In his book, Chomsky introduced the idea of generative grammar which made researchers re-think their methods and realise that they could turn to the field of Linguistics in their work.

Despite the lack of funding, research went on, focusing on semantics[1] and symbolic approaches. In addition to theoretical work being done at the time, various prototype systems were developed supporting the notion that NLP could have real-life applications.

[1] "the study of meaning in language" Sharples, Hogg, Hutchinson, Torrance and Young 1996: See 'Semantics')

Among these systems were:

- Eliza (1964-66) by Joseph Weizenbaum, a computer program designed to simulate a psychotherapist using natural language (Weizenbaum, 1966).
- SHRDLU (1968-70) by Terry Winograd at the M.I.T, which was a natural language system used by a robot to manipulate coloured blocks (Winograd, 1972).

The mid to late 80's saw a shift in trends from semantic and symbolic approaches to statistical and corpus based methods. These new methods have seen wide success with statistical based methods being used in many different NLP applications including part of speech identification and word sense disambiguation.

This project will be building on these successes using a statistical, corpus based approach to meaning rather than the traditional symbolic or linguistic based approaches used in systems such SHRDLU and Eliza.

3.2 Terms and concepts

The first stage of the proposed method involves *parsing* of the texts being compared, to find the grammatical relations between the words in each sentence. Parsing can be defined as "the process of reconstructing the derivation(s) or phrase structure tree(s) that give rise to a particular sequence of words."(Manning and Schutze 1999: pp.107).
In order to determine the relation between words in a sentence the parser has to know what is the most likely part of speech (i.e. noun, verb) of the word in the sentence. This information is acquired from a statistical analysis of a *tagged corpus* - collection of texts in which each word has been assigned its part of speech in each sentence.

Figure 3-1 shows the grammatical relation tree generated by the parsing of the sentence "He chortled in his joy":

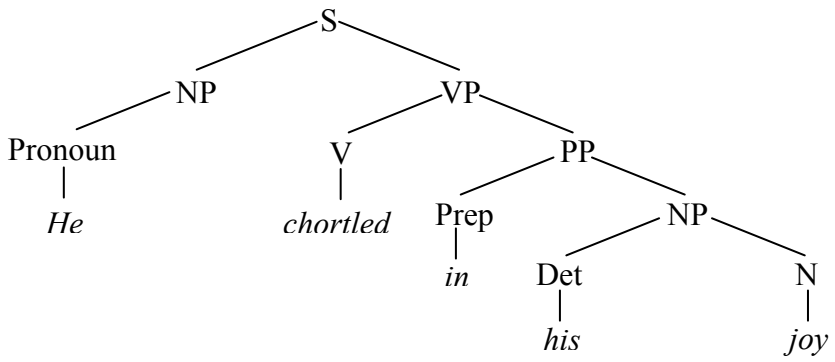

Figure 3-1: Parse tree for the sentence "He chortled in his joy" produced by the RASP parser (Briscoe and Carroll 2006: see informal description of the RASP system).

In this example the word 'in' is used as a preposition, however, according to the Chambers dictionary (Chambers 2006: see 'in') it could be used as any of 3 parts-of-speech:

1. Prep – "used to express the position of someone or something with regard to what encloses, surrounds or includes them or it; within • *Stay in bed"*
2. Adverb – "to or towards the inside; indoors • *Do come in*"
3. Adj – "inside; internal; inwards • *Never go out of the in door"*

The statistical information used by the parser lead to the conclusion that the most likely use of the word in this case is as a preposition to the noun phrase 'his joy'.

However, trees are just one way of representing sentence structures. An alternative way that has gained much popularity recently is *Dependency Analysis* which breaks-down the structure of a sentence to the way words depend on one another. Each pair of words is described in terms of the dependency type, the head and the dependent. These pairs can then be linked together to form the sentence structure. So, parsing the sentence: 'Tom saw the cat' would result in the following pairs:

subj(saw, Tom) – relation between a subject and its verbal head.

dobj(saw, cat) - relation between a verbal head and the head of the noun-phrase to its immediate right (i.e. cat).

det(cat, the) - relation between an article and the head of the noun phrase.

(Briscoe 2006: pp.17-19)

Figure 3-2 demonstrates how these pairs relate to one another in the structure of the sentence.

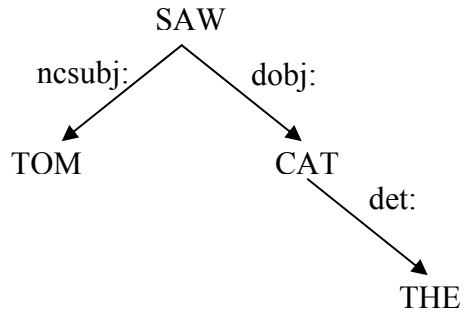

Figure3-2: Dependency grammar parse for the sentence "Tom saw the cat."

3.2.1 Word Similarity vs. Word Relatedness

When we talk about words being similar we intuitively think of synonyms, words which could potentially substitute each other in a sentence and preserve its meaning. However, there are several other ways of interpreting words to be similar. For example antonyms can be said to be very similar as they are very likely to be used in exactly the same way in a sentence even though they produce the opposite meaning. To distinguish these kinds of relationships from our classic view of word similarity we use the term *relatedness*.

This project uses the latter types of relationships to establish text similarity and therefore it would be useful to draw a clearer picture of the types of relationships it includes:

- Synonymy (look vs. see)
- Antonymy (big vs. small)
- Holonym / Meronym (hand vs. body)
    - "Y is a holonym of X if X is a part of Y".
    - *"X* is a meronym of *Y* if *X* is a part of *Y*."
- Hypernym / Hyponym (car vs. vehicle)
    - *"Y* is a hypernym of *X* if *X* is a (kind of) *Y*."
    - *"X* is a hyponym of *Y* if *X* is a (kind of) *Y*."
- Troponym (whisper vs. speak)
    - "*X* is a troponym of *Y* if *to X* is *to Y* in some manner."

(WordNet 2006: Glossary)

And the relationship it does not:

- Homonymy (river bank vs. financial bank)

3.2.2 Distributional Similarity Hypothesis

Now we have established the difference between the types of word relationships we must clarify how the similarity between words is measured. The distributional similarity hypothesis works on the assumption that words that appear in the same syntactic context are, to some degree, similar in meaning. So, for example, antonyms like 'hot' and 'cold' may not mean exactly the same thing but they are very likely to appear in a similar place in a sentence (i.e. The soup was hot/cold) and therefore can be said to be distributionally similar.

One of the most important ways of measuring distributional similarity, and the one used to collect the word data used by this system was introduced by Dekang Lin. Lin defined distributional similarity as "the amount of information contained in the commonality between the objects divided by the amount of information in the descriptions of the objects." (Lin 1998, pp.1). In other words, the similarity of two words is measured by dividing the amount of information needed to express what is common between the two words (commonality) by the sum of the information in the descriptions of each of the words separately.

$$similarity(w_1, w_2) = \frac{commonality(w_1, w_2)}{description(w_1) + description(w_2)}$$

Equation 1: Lin's Distributional Similarity Measure

3.2.3 Semantic Similarity

However, the general problem this project addresses can be seen as a test for semantic similarity between texts rather than the more usual measurement of word similarity described in section 3.2.1 or Lin's measure of distributed similarity seen above.

A sentence can be expressed in terms of a set of words or a set of the grammatical relations between the words in the sentence. An obvious limitation of using the first approach, known as the 'bag of words' model, is that each word becomes independent of the others, thus loosing any information about the structure of the sentence. The option of using the grammatical relations, acquired from the dependency parse of the sentence, overcomes this problem.

There are already several different methods for measuring semantic similarity between such sets. Manning and Schutze (Manning and Schutze 1999: pp. 299) describe 5 different measures for comparing binary vectors[2]:

- **Matching Coefficient** $|X \cap Y|$

  The number of terms appearing in both vectors.

- **Dice Coefficient** $\frac{2|X \cap Y|}{|X| + |Y|}$

  The number of terms appearing in both vectors with respect to the length of the two vectors.

- **Jaccard Coefficient** $\frac{|X \cap Y|}{|X \cup Y|}$

  As with the Dice Coefficient it measures the number of terms appearing in both vectors with respect to the length of the two vectors but also takes into account low-overlap cases by giving them a lower value.

- **Overlap Coefficient** $\frac{|X \cap Y|}{\min(|X|,|Y|)}$

  The number of terms appearing in both vectors with respect to the length of the smaller of the two vectors.

[2] Word sets with entries that are either 1 or 0.

- **Cosine** $\frac{|X \cap Y|}{\sqrt{|X| \times |Y|}}$

  Once again this measure acts as the Dice Coefficient but it penalises less if the lengths of the two vectors is very different.

The type of measure that has been applied in this project resembles the Jaccard Coefficient except that as we are concerned with matching only the entire query the division is by its size alone: $\frac{|Text \cap Query|}{|Query|}$

However, it is important to note that for this project, a similarity measure is only the starting point to be extended using lexical similarity and perhaps generalised to take into account weighted GR's.

3.3 Existing Systems

The previous sections discussed existing methods that perform similar operations to those used within the proposed method and some which are used by the system implemented to evaluate it.

This section will focus on existing question-answering systems as this is the chosen implementation to evaluate the feasibility of the proposed method. In each case the system will be described along with a short discussion of how the method used in this project differs from it.

3.3.1 Start (SynTactic Analysis using Reversible Transformations)[3]

START works by "matching syntactic structures derived from the question with those derived from natural language annotations. These annotations are machine-parseable sentences and phrases that serve as metadata to describe knowledge segments" (Katz, Felshin, Yuret, Ibrahim, Lin, Marton, McFarland and Temelkuran 2002: pp. 3).

It uses Omnibase, a specially comprised virtual database "that integrates heterogeneous data sources using an *object–property–value* model." (Katz, Felshin, Yuret, Ibrahim, Lin, Marton, McFarland and Temelkuran 2002: Abstract) to access different kinds of information in a uniform way. The object-property-value model Katz et-al refer to is based on the idea that many questions can be answered by extracting the object of the questions, a property of the object the question is looking for and the value for that property in a data source. So, in the question 'What is the capital of China?' the object would be China, its wanted property would be its capital and a data source should hold 'Beijing' as the value.

The creators openly state that the system is only designed to deal with questions of a possessive type or ones using an 'of' relation (i.e. What is the capital *of* China) but justify this by claiming that object–property–value type questions have been found to occur quite frequently.

[3] (Katz, Borchardt, Felshin, Et-Al: Homepage)

One of START's greatest appeals is its full use of natural language in its interface both interpreting a naturally phrased question and responding with a relevant answer in natural language. In addition, it is highly likely that if an answer is provided it would contain the exact information searched for because of START's use of the object-property-value model. However, as demonstrated in Figure 3-3, its main limitation seems to be that it cannot deal with different ways of phrasing the same question.

(a)

(b)

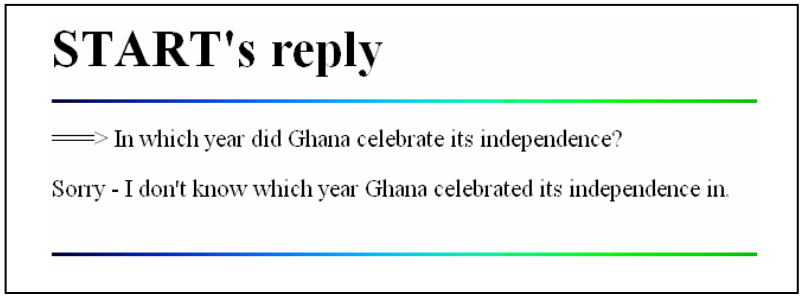

Figure 3-3: START's response to:
(a) 'When did Ghana get independence?'
(b) 'In which year did Ghana celebrate its independence ?'

The use of grammatical relations in the proposed method should mean that a system using it would be able to deal with the different ways of asking the same question.

3.3.2 MULDER

MULDER "relies on multiple search-engine queries, natural-language parsing, and a novel voting procedure to yield reliable answers coupled with high recall" (Kwok, Etzioni and Weld 2006: Abstract).
The process MULDER goes through involves several stages. It starts by passing the natural language question inserted by the user to a Parser (as described in Section 3.2) to produce the question phrasal structure (Question Parsing stage). This information is then used by what Kwok, Etzioni and Weld call the Question Classifier stage to extract the information required in order to formulate search engine queries (Query Formulation stage). The web pages resulting from these queries are scanned for relevant sections (Answer Extraction stage) which are in turn scored and ranked (Answer Selection stage) before being presented to the user (Kwok, Etzioni and Weld 2006: pp. 4).
The different stages in its search process are a good way of taking different aspects of search into account: i.e. retrieving relevant documents before searching the content for a precise answer. This narrows the computation time and the chance of false answers.

The proposed system is in fact better compared to the answer extraction and selection stages rather than the entire process MULDER goes through. The two stages start by using keywords to find and then rank sections which could contain the answer. These sections are then clustered together into keyword similarity groups, the average rank of which is then found. The highest ranking answer of the highest ranking group is returned as the result. This process is entirely different to that of the proposed method which uses a comparison of the grammatical relations within a sentence taking into account word neighbour data.

In its Question Classifier stage however, MULDER uses the relation between words to find the object of the verb which they use to determine what type of question it is (and hence what type of answer is required). This same piece of information, the relation between words, is also used by the proposed method but as a base for comparison between sentences rather than a way to extract information (more detailed use is described in the Design, Section 5).

MULDER uses a much more precise handling of situations than the proposed system, as at most of the stages it takes different situations and word combinations into account. This means that it is not very flexible, translating the system into another language for example, would inevitably require much development work to fit it to the rules of the new language. However, its extended stage process means that its results are likely to have a high precision and its use of search engine technology means that it can easily (and relatively cheaply) be run on a very large data source such as the web.

3.3.3 Google Q&A[4]

"Google's Q&A uses open Web resources, not proprietary information or subscription databases, to answer questions." says Peter Norvig, Google's Director of Research, "We look through our logs, do research on question answering, look at question forms, and try to come up with as many variations as we can." (Tomaiuolo 2005).

In a talk at UC Berkeley (Chitu 2006) Norvig explained that Google Q&A uses an automated search for patterns based on examples, as a method of extracting facts from a variety of resources. He explained that the process starts by looking at relations between facts to find these patterns. As an example he gives the pairs (Bob Dylan, 1941) and (Vincezo Bellini, 1801) which are looked up in a corpus to see where they occur. Table 3-1 demonstrates how the sentence: "Vincezo Bellini was born in Cantonia in 1801" is analysed to produce a pattern. These patterns are then generalised to produce more global ones in which fact types such as year and city name are incorporated.

Prefix	Arg1	Infix	Arg2	Postfix
NULL	V.B.	was born in Cantonia in	1801	.

Table 3-1: Google Q&A process of finding patters

The idea behind Google Q&A slightly resembles the START system in its use of the pattern in which information is available, however, its advantage is in the fact that these patterns are automatically generated from examples rather than relying on pre-set rules.

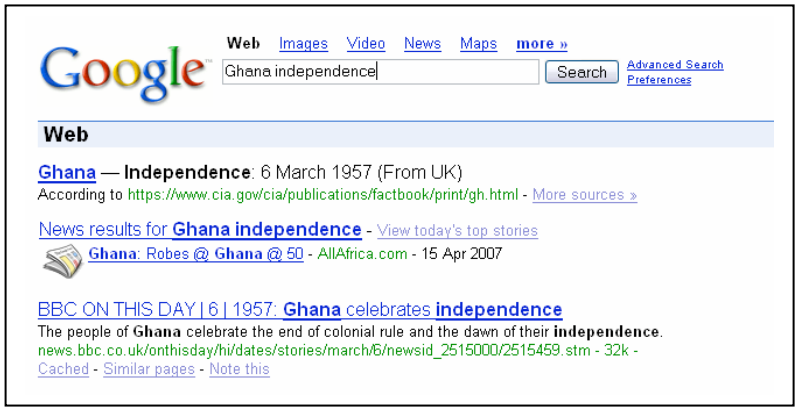

Figure 3-4: Google Q&A keyword results

[4] (Google 2007a).

But, despite what seems to be a process allowing for high flexibility, the resulting system still uses keywords as its input rather than natural language questions. It seems that the system does not take well enough account the possible patterns used to search for the information.

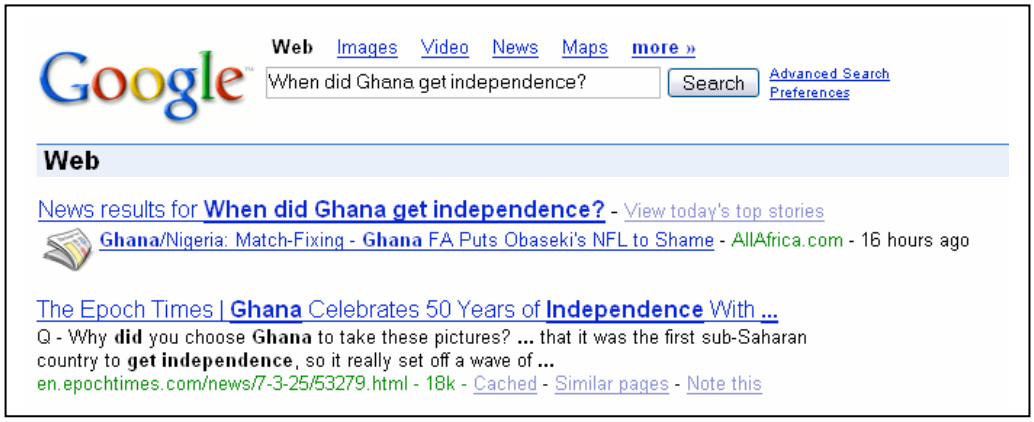

Figure 3-5: Google Q&A question results

3.3.4 Database Management Systems (DBMS)

Another group of systems which currently perform a similar task are database management systems (DBMS) which respond to a query inputted by the user by providing the relevant information from the database. These queries use a pre-defined query structure and keywords which the DBMS then picks out and performs actions accordingly.

This makes these systems limited to the domain of databases in general, meaning they could not be applied directly to a collection of texts such as the web. In addition, they are also limited to the specific domain of the database being queried, so a query being run on one system (say, holding information about films) would not necessarily work for another (such as a recipes database) because of the difference in structure between the two systems.

4. Requirements

Using natural language to find answers is the most intuitive method for us. When we search for information we want to be able to ask a question and be answered as if we are talking with a person with a vast amount of knowledge.
Currently the most common technique for providing a user with the information he is looking for is Information Retrieval, queries resulting in relevant documents rather than a precise answer. In addition to the imprecise result, this method requires the user to possess knowledge of which keywords are likely to get him more relevant answers. Even people who use this method very frequently find that it may take several attempts before determining a suitable set of keywords. This is yet another example of how the human user has to adapt himself to make use of the computer instead of the other way round.

One way of approaching this problem is to imagine it being part of a "super" search engine. One that would take an input in natural language, search through the World Wide Web and return a correct or appropriate response again in natural language. This project will attempt to develop the search methodology for such a system.

As this is a research based project, its main requirement it to establish whether the suggested method is valid and to find its advantages and limitations. For the system demonstrating the method to be entirely successful it would have to return all and only those sentences which answer the query with what is deemed to be a relevant answer. To allow for some flexibility, this requirement can be rephrased to specify as few as possible irrelevant answers and as many as possible of the available relevant ones.

At this point it would be wise to clarify what is meant by a relevant answer. As there is by no means a clear-cut definition for what is a relevant response to a query, this will have to be a subjective decision made by the person posing the query. To increase objectivity of this test, impartial participants will be asked to pose some questions and rate the responses they receive both from this system and from an equivalent system using a different method to achieve the same task (this will be discussed further in the Evaluation section).

5. Design

5.1 Design overview

The core part of this project focuses on information retrieval, setting out to retrieve a set of relevant sentences as a response to a query given in natural language. This is achieved by a 4 stage process (further details in Section 5.2) as represented in Figure5-1:

1. **Parsing** – Taking the Query and the Base Texts and parsing them to sets of grammatical relations (from now on referred to as GRs).

2. **Filtering** – Picking out only the relevant information from the parser output to remove irrelevant or too finer grained information.

3. **Comparing** – Computing the similarity of a base text sentence to the query sentence by comparing the GR sets making up each of the sentences. This is achieved by:
    a. Finding the GRs in the text that have the same structure to a GR in the query.
    b. Extending each component of those GRs to a set made up of all its neighbour words to account for word similarity (doing the same to the query GR).
    c. Counting the overlap between the new extended sets.
    d. Summing the overlap count for each of the text sentence GRs to get its total similarity score.

4. **Output** – Returning the sentences with the most overlapping to the query sentence.

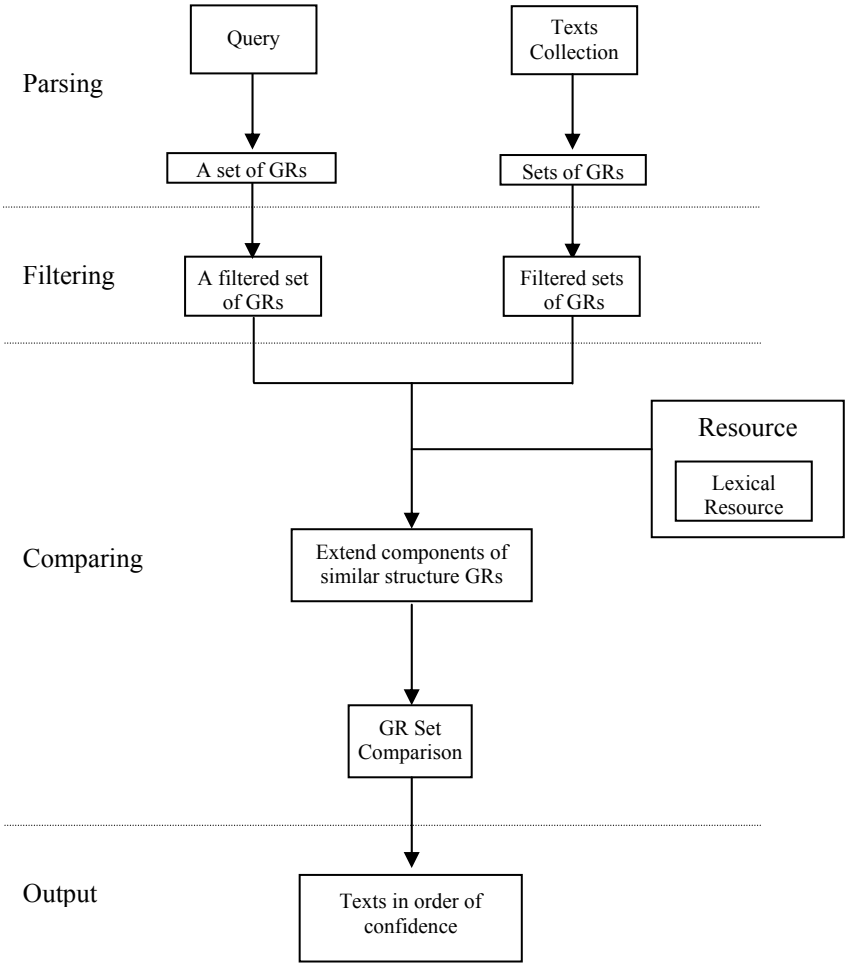

Figure 5-1: My Method Process

To further clarify the suggested process described above, Figure 5-2 on the next page demonstrates the different transformations a query and a collection of texts go through during the process. In this example the user inputs the query: "What did Tom observe?". This, along with the texts, is parsed to sets of grammatical relations, which are then filtered ready to be compared.

In the example there are 2 texts in the collection, each producing a single set of GRs. The sets of Venn diagrams show how only the GRs with a structure matching to one in the query are expanded to include the neighbour data. Each of the text sets is then compared to each of the sets in the query to find their intercept.

This example also demonstrates that despite the query using the term 'observe' and the text using the term 'see' the system would still be able to match them at the lexical comparison level.

1. Tom saw the cat. (High confidence)

2. Tom ate lunch. (No match)

Figure 5-2: Process Example

5.2 Detailed Design

5.2.1 Parsing

Before any comparison can be performed, the system has to get the full information about the sentence structure. This is achieved using the RASP parser (Briscoe, Carroll and Watson 2006) which takes a text as input and returns the grammatical relations it is made of. RASP parses the text in a 4 stage process (Briscoe and Carroll 2006):

1. **Tokenisation** - marks sentence boundaries and separates punctuation from words they are attached to.
2. **Part of speech tagging** - each word is tagged with it's most probable part of speech for the sentence it is in (i.e. talked: V)
3. **Lemmatization** - the stem of each word is found and separated from its inflection (i.e. talk+ed).
4. **Parsing** - analyses the part of speech tags and generates a parse tree representation of each sentence.

The result is a collection of relations made up of the relation type descriptor (see Briscoe 2006 for details of relation types) and the words the relation applies to in the form: word root, its inflection, its occurrence in the sentence and its precise part of speech (i.e. (|ncsubj| |eat+ed:2_VVD| |Tom:1_NP1|).

5.2.2 Filtering

The Filtering stage dismantles the set of GRs produced by the RASP parser into its components and retains only the necessary information. This information is then stored in a purpose built Data-structure (more information about the data-structures used can be found in Implementation, Section 6).

For the purpose of the implemented system the following types of information is discarded:

- Relation types such as Auxiliaries (between a main verb and a dependant, i.e. "should eat") and Determiner (between a noun and an article, quantifier, etc. i.e. "the dog") (Briscoe 2006: pp. 17).

- Some information about each word such as its place in the sentence, its inflection and its exact part of speech description (i.e. noun instead of plural common noun (UCREL 1993-2007)).

In addition, relations which are found to be passive are 'flipped' to their active form (i.e. from Tom was seen Mary to Mary saw Tom).

5.2.3 Comparing

As mentioned in the design overview, the comparison is performed at two levels. First the structure of the GR is examined and only those in the text that have the same structure to a GR in the query are passed to the second, word similarity, level.
The GR structure is compared by looking at the relation type descriptor, if they do not match, the similarity score between the two GRs is set to 0.
The word similarity comparison is applied to each of the words making up the relation passed to the second level. Each word is looked up in a collection of word neighbour data[5], which holds for every root word, a set number of related words each with a score of how similar they are to the root word. The relation words are expanded to include their neighbour data and then the relations as a whole are compared.

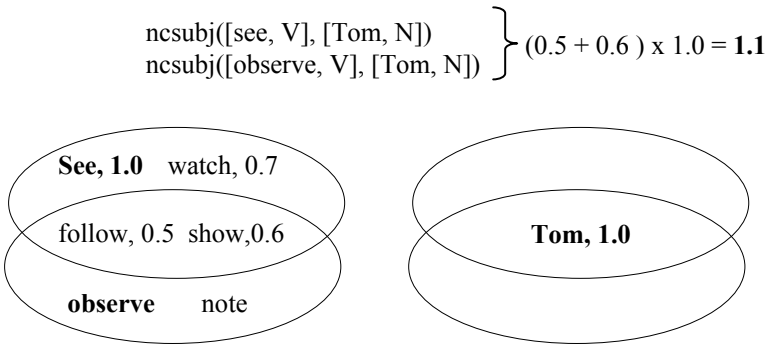

Figure 5-3: Word similarity comparison

This is done by first checking if the pair of words at the same place in the relation have the same part of speech and if so, finding the overlap between their neighbour sets.

[5] I would like to thank Diana McCarty who provided me with this data which she collected using the Lin distributional similarity measure described in section 3.2.2 (see Appendix IV for Neighbour Data Samples).

The calculation of the neighbour set overlap score can be achieved by one of two ways:

a. Counting the number of words shared by the two sets.
b. Summing up the neighbour similarity score of each of the shared words to the query root word.

The results for each of the places in the relation are then multiplied to get the total similarity between the two relations, and the sentence similarity score is the sum of the scores of the GRs it is made up of.

5.3.4 Output

As the comparison is performed, all sentences with a positive total similarity score are collected. When all the sentences in all the texts in the database have been compared the sentences are retuned with the highest matching sentences first.

6. Implementation

Since the system is implemented as a demonstration of one of the ways the proposed method can be used, it was designed keeping the option of extending it to other applications in mind. In addition, as it involves processing and then using a large amount of data it was very important to keep memory usage to a minimum. This involved creating scripts and classes for pre-processing data intended to be run prior to the routine use of the system.

Another decision based on the fact that the system is only implemented as proof of concept was to keep the development of its usability to a minimum. Therefore, the system has no graphical interface and currently only runs from the command line of the University of Sussex teaching server.

The bulk of the code was written using the Java programming language mainly due to my prior familiarity with it. However, since the system had to interact with the Rasp system running on a separate Unix machine some shell scripts were used to bound the two processes together.

6.1 Data Structures

In order to increase the readability of the code and to make it easily extendible four specially designed data-structures were created:

6.1.1 Relation

Represents the information making up a grammatical relation returned by the Rasp parser. A single relation is made up of three pieces of information:

- A collection holding one or more relation type descriptors (i.e. ncsubj, obj)
- A collection holding one or more words each with their respective data in an Array (i.e.[can, V], [dog, N]) - which the relation applies to
- An array holding the score of the relation's similarity to each of the query relations.

6.1.2 Sentence

Contains the value of a sentence in different representation forms:

- As a string of plain text: Tom is tired
- A collection of arrays holding the words of the sentence each with the data describing it such as its part of speech and inflection: [Tom, N], [be, s, V], [tire, ed, V].
- As a collection of Relation objects: [ncsubj] , [[tire, ed, V], [Tom, N]];
  [aux] , [[tire, ed, V], [be, s, V]];

6.1.3 Result

Used for keeping record of, and giving easy access to, all information relating to a sentence which is to be returned by the system as a possible result.
This data structure provides the information relating to the origin of the sentence:

- The name of the text it came from
- It's location within the text (line number)

Along with the sentence object itself and its similarity score to the current query.

6.1.4 Pair

Holds the information making up a Pascal textual entailment challenge pair. Each holds an id, its real entailment value and the similarity score given to it by the comparison method, along with the text and hypothesis in plain text and as a Sentence object.

6.2 Pre-Processing

There were two collections of information which could potentially make the processing time very slow: the database of texts and the neighbour data. To avoid slowing down the application, scripts and classes were written to perform as much as possible of the processing ahead of running the system as a user end application.

6.2.1 Preparing the Texts

Preparing the texts involved passing them through the Rasp parser and then transforming the output into a filtered collection of Sentence and Relation objects for quick and easy access by the comparator.
This was achieved by writing a shell script to call Rasp on each of the texts and then running a java class which performed the filtering process on each of the Rasp output files.

6.2.2 Neighbour Data

The source of the word-neighbour data (as mentioned in Section 5.2.3) came in the form of four text files for nouns, verbs, adjectives and adverbs. Each of these files contained a hundred neighbour words for each root word in the structure:
<root word> <neighbour word> <similarity score>.
Using the entire available collection would have made the system very slow and was not necessary in order to get an idea of the similarity between two words. Therefore, the data was trimmed to include a small pre-set number of neighbour data for each root word.
In addition, as with the database of texts, time could be saved by using a pre-created object for quick and easy access during the comparison process. In this respect, the Neighbour class is also used as a data-structure providing methods for accessing the neighbour data.

6.3 Filtering

As demonstrated in Section 5, the comparison process is divided into four stages, the last three of which had to be specifically implemented for this project. Stage 2, the filtering, which deals with the transformation of the texts and queries into a structure that can be used for the comparison, is performed by classes in the Filtering package. Stages 3&4 comparing and returning the best matching sentences are performed by the Comparator package.

The filtering package is design to be easily extended for the use of other applications working with the output of the Rasp parser. The RaspLineTokeniser class therefore, simply transforms a text file created by Rasp into a collection of Sentence and Relation objects, while the ProjectIO class provides static file access methods. The Filter class is the only one in the package built solely for the purpose of the current application.

The filtering process is initiated by the Filter class which calls RaspLineTokeniser to create the Sentence and Relation objects. These objects are then returned to the Filter object to apply the application specific filtering on them (see section 5.2.2 for details of the filtering applied in this application).
Figure 6-1 demonstrates the interaction between the classes in the Filtering package.

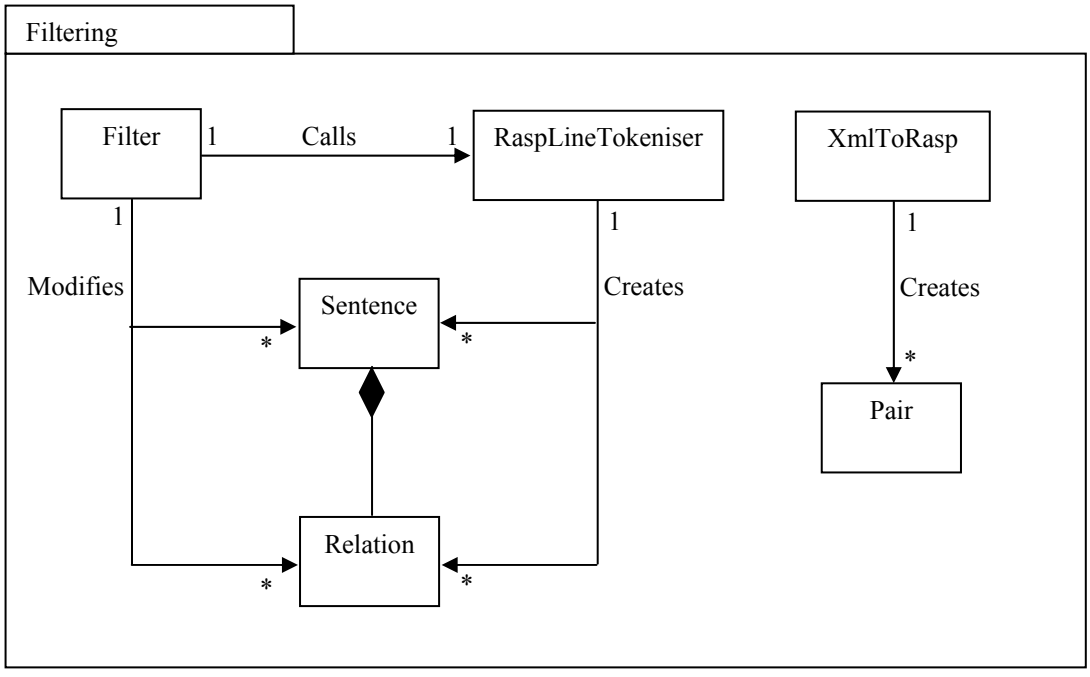

Figure 6-1: Filtering package class diagram

A later, unrelated addition to this package are the two classes designed to prepare the evaluation of the method using the Pascal textual entailment challenge data-set (see Section 7.1 for details of this experiment). The XmlToRasp class reads the data-set, creates Pair objects and prints their information into a text file Rasp can parse.

6.4 Comparator

The main purpose of the Comparator class is to use the proposed method to compare between texts in the database and a query entered by the user. However, it also performs another type of comparison method used as a base line to evaluate the system output and a comparison on the Pascal textual entailment data-set using the proposed method.
The main class of the package is the ComparisonController which calls the comparison method class on all the texts, stores and finally returns the resulting sentences.
As the same process had to be performed using two different methods, an abstract class was created to be used by the ComparisonCotroller. This is the Comparator class which requires the query as a Sentence object on initialisation and provides a compare() method which gets another Sentence object as its input and returns the similarity score to the Comparator object query.

The two classes extending the Comparator abstract class are:

- **SetCompare** – The class performing the set comparison method which the project proposes. It performs the comparison as described in Section 5.2.3.
- **WordCountCompare** - A class performing the chosen alternative comparison method used to evaluate the project results. The comparison is done by counting the number of occurrences of each of the query words in the sentence using the tagged text format stored in a Sentence object.

Figure 6-2 demonstrates the interaction between the classes in the Comparator package.

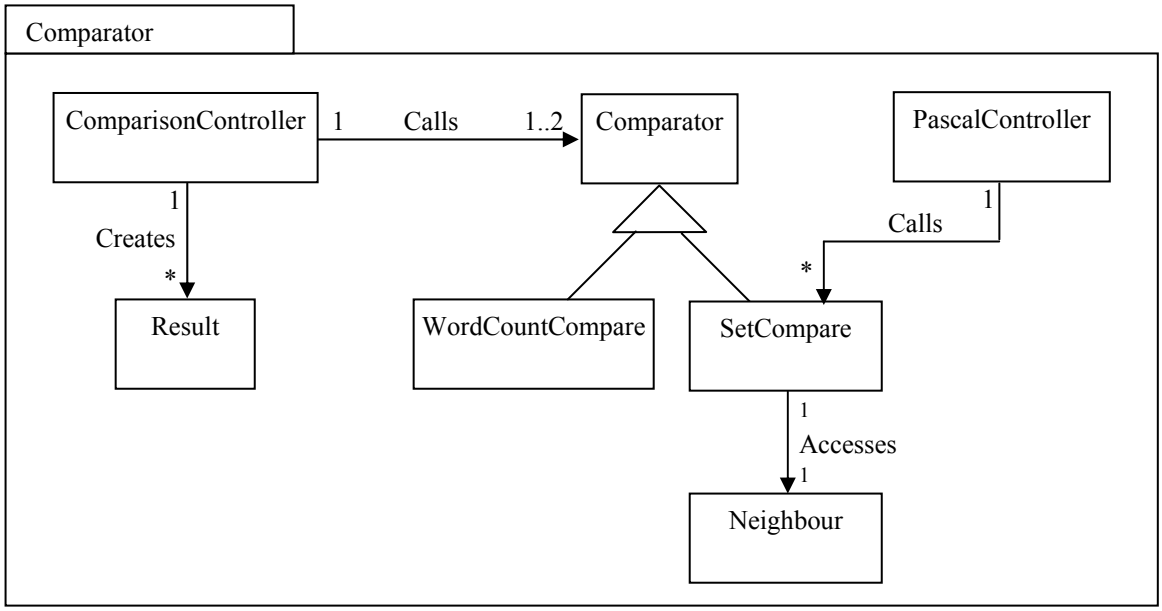

Figure 6-2: Comparator package class diagram

As in the Filtering package, here too a class was later added in order to perform the Pascal textual entailment test. The PascalController class behaves exactly the same as the ComparisonController except that it does not require the use of the Comparator interface.

6.5 Overall structure

To connect between the java application and the RASP parser for the execution of the question answering system a short script was written. The script accepts a sentence as its argument and passes it to the Rasp parser which stores the result in a text file. The script then calls the java application which opens the text file created by Rasp and performs the comparison on the parsed sentence.

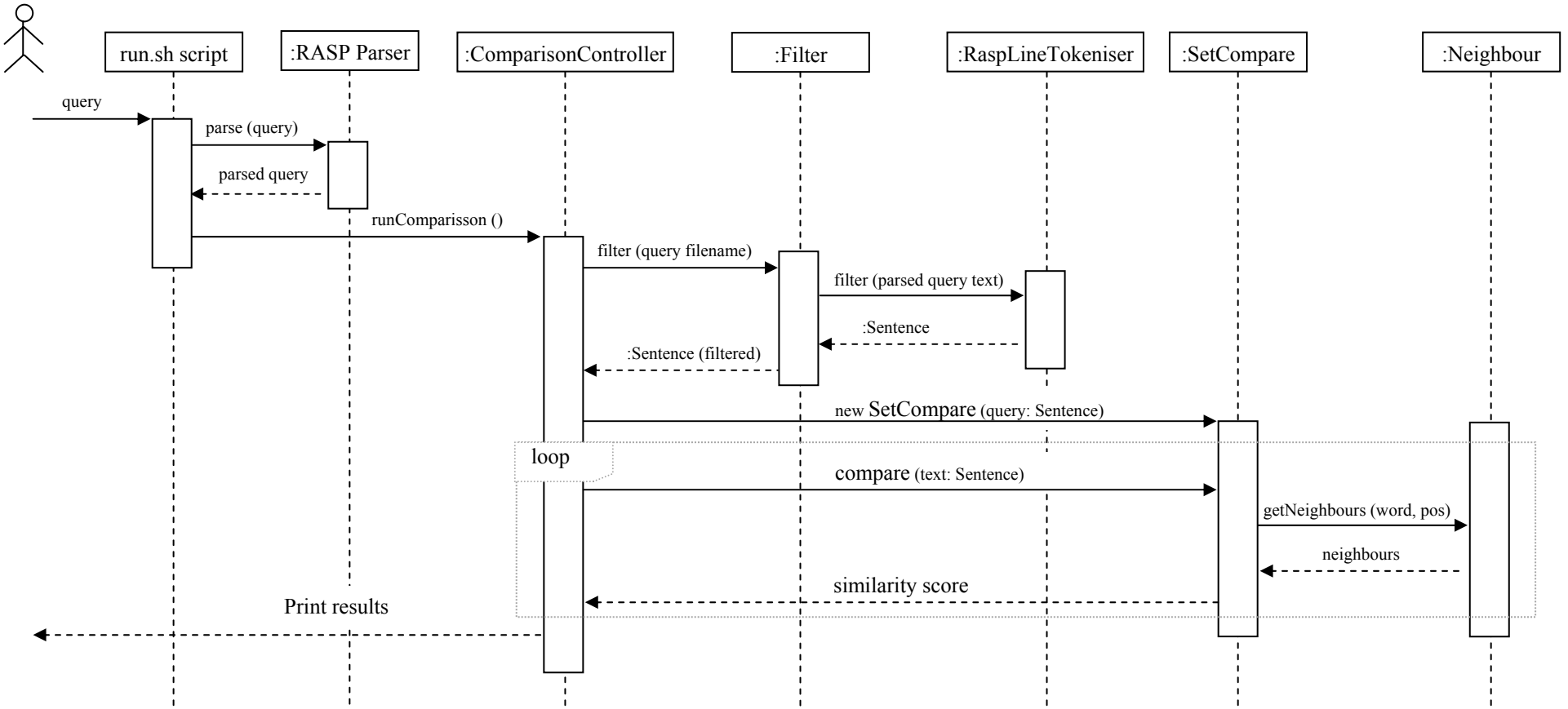

Figure 6-3: Sequence diagram of the set comparison process

7. Evaluation

The evaluation of the output of the question answering system is an integral part of this project. The quality of the output reflects on the feasibility of the proposed method and demonstrates its advantages and weaknesses. In order to know that the interpretation of the output is not biased, and whether the method could be applied to other NLP tasks several types of experiments were conducted.

7.1 Experiments

1. Absolute

   Several texts about certain subjects[6] were downloaded from the web and parsed through the Rasp and filter systems to act as the database of texts on which the system will run[7]. Two types of tests were then performed on these texts:

   a. Subjective

      Questions of various forms were written to test how the system handles different query structures.

   b. Objective

      Impartial participants were asked to pose questions which they would expect to be answered using the provided texts and then rate the top 20 results on a relevancy scale.

2. Comparative

   The same tests exactly[8] were run on a "Google-like" search engine as a baseline to which the system results can be compared to.

3. Textual Entailment

   The PASCAL Textual Entailment Challenge "was proposed … as a generic task that captures major semantic inference needs across many natural language processing applications"(Dagan, Glickman and Magnini 2006: Homepage).

[6] Each subject was covered by at least two sources in order to produce redundancy of information to some extent.
[7] These texts were manually collected and therefore, not filtered to remove heading and menu information.
[8] Using the same questions written for the Subjective and Objective tests.

This test acted as another objective assessment of the systems potential with respect to the method's potential use in applications other than question answering.

The test was run on most of the development data-set provided for the first Recognising Textual Entailment Challenge[9]. This data took the form of xml 'pair' tags each holding a text and an hypothesis for which the system had to determine whether they matched or not. The results of running the system on the pairs was evaluated by using the 'value' tag attached to each pair which indicates whether they match or not and comparing it to the similarity score provided by the system. A result was deemed correct if the pairs *value* was set to TRUE and the similarity score > 0, or if the value was FALSE and the similarity score = 0.

[9] Downloaded from http://www.pascal-network.org/Challenges/RTE/Datasets/ (Available 19 April 2007).

7.2. Evaluating the Results

7.2.1 Evaluation Techniques

There are many ways in which we could evaluate the output of an NLP system, the most instinctive is by simply counting how many of the retrieved answers are correct and how many are not. However, this measure, known as *accuracy* and *error*, does not provide us with the most useful information. If, for example, a system provides us with no results at all, we could say that that it made no error, even though common sense tells us otherwise.

The solution to the problem described above, and to other similar situations, is to measure the output of a system with respect to the number of possible correct answers, the total number of answers returned by the system, and by the number of returned answers which are correct. The two main measures using this information are called *precision* and *recall*.

Precision – how many of the answers returned by the system were judged to be correct.

Equation 7-1: Precision

$$\frac{Correct \cap \mathrm{Re}turned}{\mathrm{Re}turned}$$

Recall – how many of the correct answers possible, were found by the system

Equation 7-2: Recall

$$\frac{Correct \cap \mathrm{Re}turned}{Correct}$$

For a system such as the one implemented in this project to perform best there needs to be some kind of balance. For example, the system could return 1 answer, and if this answer is judged to be correct, it would mean the system has 100% precision. However, if the total number of possible correct answers is 10 its recall is very low. On the flip side, a system could get 100% recall by returning all possible answers, among them all the correct ones, but this would leave it with a very low precision value.

In order to find a single way to measure overall performance of a system, taking both precision and recall into account, the *F-measure* was introduced. This formula uses an added factor, β, to determine if precision or recall should be favoured and by how much.

Equation 7-3: F-measure $$\frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

The most common use of the F-measure equation, and the one used here, is when β is set to 1 giving precision and recall equal weight. This can be simplified to:

Equation 7-4: Simplified F-measure $$\frac{2PR}{P + R}$$

7.3.2 Experiment Results

As mentioned in the design, Section 5.2.3, there are two possible ways of calculating the overlap between the word neighbour sets, by counting the number of overlapping words or by summing up the similarity scores of each of the overlapping words to the query word. Running the absolute tests using both methods revealed that the latter produces better results (see Appendix I, page 65) and therefore, all results quoted below were produced using this method. The full test results are available in Appendix I.

Subjective tests

The results of the subjective tests using both the proposed (sets) and the base-line (words) methods are demonstrated in Table 7-1. The average total number of answers returned by each method for each question is shown, as well as the percentage of relevant results with top rank and the average rank of the top relevant result.
Precision and recall figures are given both with respect to the top 20 answers returned by the systems, and with respect to the top 5 answers.

	Sets	Words
Average number of results	11.60	457.33
Relevant results in first place	33.33%	13.33%
Average index of top relevant result	4.40	5.46
Average relevant results in top 20	0.93	1.40
Precision [10]	8.60%	7.00%
Recall	35.78%	54.00%
F-measure	13.87%	12.39%
Average relevant results in top 5	0.47	0.87
Precision	5.74%	4.33%
Recall	24.89%	37.22%
F-measure	9.33%	7.76%

Table 7-1: Summary of subjective results using both methods

[10] Precision was calculated by total number of results *after* trimming to a maximum of 20.

Table 7-2 shows the results of posing the same question using different wording, one with the actual word used in the text and another with a related word. For each question the number of possible correct answers assessed by manually reading through the texts is provided, along with the total number of answers given by each method and the index of the top correct result. Precision and recall figures are again shown in terms of the top 20 and top 5 answers returned.

Question	**Was the Apollo 11 moon landing faked by NASA ?**		**Was the Apollo 11 moon landing counterfeited by NASA ?**	
Potential Correct answers	5		5	
	Sets	**Words**	**Sets**	**Words**
Number of results returned	15	912	15	912
Index of top relevant result	1	3	1	5
Relevant results in top 20	2	3	2	2
Precision [9]	13.33%	15.00%	13.33%	10.00%
Recall	40.00%	60.00%	40.00%	40.00%
F-measure	20.00%	24.00%	20.00%	16.00%
Relevant results in top 5	1	1	1	1
Precision [9]	6.67%	5.00%	6.67%	5.00%
Recall	20.00%	20.00%	20.00%	20.00%
F-measure	10.00%	8.00%	10.00%	8.00%

Table 7-2: Results for the same question using different wording

Figure 7-1 in the next page is an example of the top 5 results returned by the sets method for a given question. The information provided with the answers to a question is the total count of sentences found[11], and for each answer it's similarity score to the question and its location in the database of texts (i.e. text name and line number in the text).

One very noticeable aspect of the quality of the results returned by either systems, demonstrated in Figure 7-1, is the large amount of irrelevant data returned. On closer look it can be observed that these irrelevant answers would, in many cases, not be considered acceptable answers to any question as they consist of headers and menu items. Another frequent type of irrelevant answer is one from an entirely different subject text as can be seen in answer number 2 of Figure 7-1.

[11] As they are trimmed to the top 20 answers when returned as a text file and top 5 when printed to screen.

Query: When did the Berlin Wall fall ?

```
There are 11 matches. The best matches in order are:

1. Score: 6.0     Text: Berlin_Wall-wikipedia.txt     Sentence 0:
Berlin Wall From Wikipedia , the free encyclopedia ( Redirected from
Berlin wall ) Jump to : navigation , search East German construction
workers building the Berlin Wall , 20 November 1961 .

2. Score: 2.6666666666666665  Text: French_Revolution-about.txt
      Sentence 4:
A few histories stop in 1795 with the creation of the Directory ,
some stop in 1799 with the creation of the Consulate , while many
more stop in 1802 when Napoleon Bonaparte became Consul for Life or
1804 when he became Emperor .

3. Score: 2.0     Text: Battle_of_Hastings-bbc.txt    Sentence 39:
As the day drags on , the numbers began to tell and the English
shield wall begins to crack. * Late in the day , Harold takes an
arrow in the eye and as his men mill around him , four Norman knights
break through and hack him down .

4. Score: 2.0     Text: Berlin_Wall-about.txt   Sentence 33:
At least 100 people were killed at the Berlin Wall , the last of them
was Chris Gueffroy ( 1989-02-06 ) .

5. Score: 2.0     Text: Berlin_Wall-about.txt   Sentence 43:
On February 1997 , a red line was painted on the pavement at the
former \ Checkpoint Charlie \ to mark the course of the former Berlin
wall .
```

Figure 7-1: Irrelevant results

Objective tests

Table 7-3 provides the summary of results returned for the subjective tests by both methods, it contains the equivalent information to that in Table 7-1. For each method the average number of answers returned is shown along with the number of top ranking answers and the average index of the best ranking answer. Recall figures are not provided for these tests as to have the participants determine the number of possible correct results was deemed too time consuming. Precision figures are again provided with respect to the top 20 and top 5 answers returned.

	Sets	Words
Average number of results	12.35	20.00
Number of relevant results in first place	15.00%	10.00%
Average index of top relevant result	4.00	5.50
Average relevant results in top 20	0.30	0.75
Precision	2.20%	3.75%
Average relevant results in top 5	0.20	0.45
Precision	1.70%	2.25%

Table 7-3: Summary of objective results of proposed method

Figure 7-2 shows the distribution of grades given by the objective participants to the results returned by the two methods. These results highlighted the fact that both methods find a substantially higher number of irrelevant answers to relevant ones, with the results of both methods containing 83% of what the participants deemed to be bad answers.

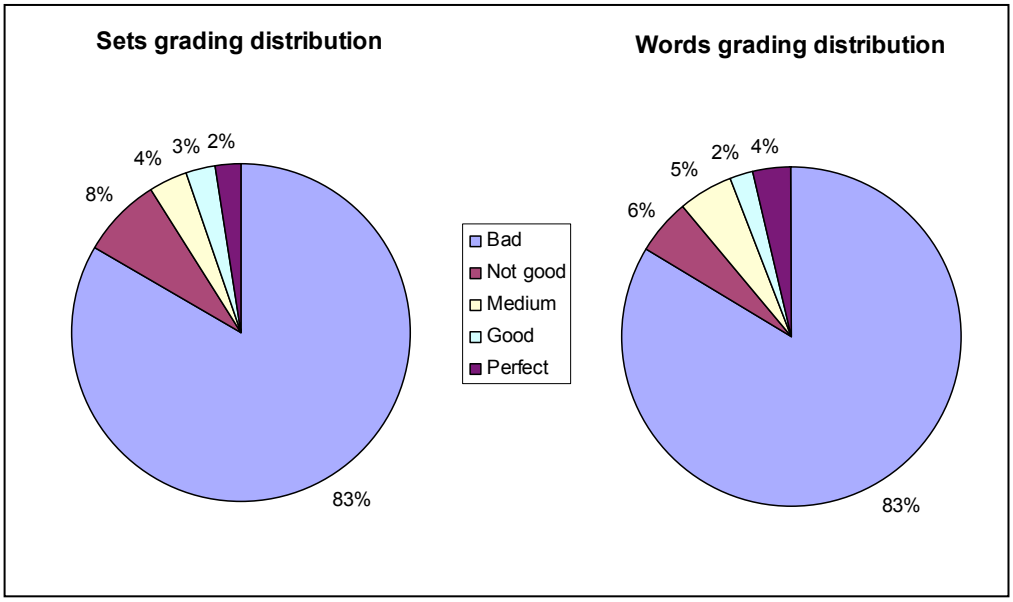

Figure 7-2: Grading of proposed method results by objective participants

Textual Entailment

Running the proposed method on the Pascal Textual Entailment Challenge data-set produced a very low percentage of correct results as indicated in Table 7-4 which gives the proportion of correctly and wrongly classified answers. The possible reasons for this are discussed in the next section.

Correct	50.90%
Wrong	49.10%

Table 7-4: Summary of PASCAL challenge data results [12] [13]

[12] Where a result is considered correct if the value given to it in the source data is TRUE and the similarity score > 0, or if the value is FALSE and the similarity score = 0.
[13] The comparison was only run on pairs with 1 sentence in the text.

7.4 Analysis

In general the results of all the tests showed that there is still work to be done in order to sufficiently improve the output of the method to make it a viable solution to the question answering tasks. However, the results have also highlighted some key characteristics of the method which can be taken into account when considering whether to use it and how it should be implemented.

From the onset, it was assumed that this method would perform better given a large mass of data which is likely to have high redundancy. When dealing with a large amount of data, precision, rather than recall, becomes the more important measure of success. This is due to the fact that in any very large collection of data there is inevitably going to be a high level of redundancy making the number of possible relevant answers to a question immense.
Thinking of a search performed on the internet, one would prefer to be given a smaller number of very relevant results to a very large list that has to be sifted through. Unfortunately, due to limited resources, the system was only tested on a relatively small amount of data in which the redundancy factor was reproduced to a limited scale. Despite this, the hypothesis of the benefit of redundancy to this method has been supported by the fact that the results pointed to the precision of the method as its strong point.
The results given in Table 7-1 showed that the proposed method returned a much more reasonable amount of answers than the base-line method, averaging at about 11 returned sentences per question vs. the hundreds returned by the word count method. This was also reflected by the slightly higher precision and f-measure values which were calculated using the top 20 and top 5 results only[14]. Furthermore, the ranking of the relevant results was slightly higher in the proposed method results with 33% of the questions getting a relevant answer in first place as opposed to 13% using the base-line method.

As stated above, the precision, recall and f-measure values were calculated using a trimmed section of the actual outputs returned. The average number of results retrieved shown in Table 7-1, demonstrate that in some sections this trimmed section was only a

[14] Note that had the Words method output been evaluated using the entire set of returned results it would not have been possible to efficiently determine the total number of *relevant* results it returned as the total was usually in the hundreds.

minor part of the entire collection. As most of the tests performed using the proposed method resulted in quite a small number of results, the precision figure stated is an adequate representation. However, had the entire collection of results been used for the base-line method, the difference between the two values would have no doubt increased substantially.

Recall seems to be the weak point of the proposed method, this can be observed by the fact that even when only looking at the top 5 results of each method, the recall value for the baseline method is about 10% higher. This may be a compromise that will have to be made to achieve the high precision aimed at, and indeed should become less important when the method is run on a larger collection of data.

As Figure 7-1 demonstrated, many of the results were in fact menu items or headings and therefore could not be accepted as relevant answers. This occurred due to the fact that the texts used were manually collected and as such, not filtered to remove information of that sort. Obviously this influenced the quality of the data to some negative extent, but, as both methods were run on the same texts, this did not affect the comparative results. In addition, this problem could easily be rectified by employing an external application to withdraw only the relevant information from web pages and thus does not lessen from the method's validity.
An additional problem noted with the quality of the results is that of finding ones from a text about a different subject to the question posed. As the system was run on a sentence by sentence basis, and due to the similar nature of the texts collected (historical events), this situation is not un-expected. Consequently, this problem should lessen greatly when running the system on a larger collection of data due to the nature of redundancy discussed above. In addition, as with the previous problem, this situation can be easily rectified by combing the method with a document or website tagging system such as the one used by Google.

The objective tests results supported the trend noted in the subjective tests in which the proposed method returned many less results than the word count method. The new information revealed in these tests (see Table 7-3) was the reduced precision achieved by the proposed method in comparison with the baseline. This fact demonstrates the

importance of running objective tests in addition to subjective ones, however, the very minor difference between the precision figures indicates that this is not a crucial problem.

Finally, the results of the tests run on the Textual Entailment Challenge data-set show that it acted no better than a random baseline would have. When analysing this result one has to take into account the fact that the system was not designed specifically with the challenge in mind. In particular, the challenge sets-out to include different types of texts and relations between the text and hypothesis pairs.

```
<pair id="56" value="TRUE" task="IR">
      <t>Euro-Scandinavian media cheer Denmark v Sweden draw.</t>
      <h>Denmark and Sweden tie.</h>
</pair>

Euro-Scandinavian media cheer Denmark v Sweden draw.
(|ncsubj| |v:5_VVD| |Denmark:4_NP1| _)
(|dobj| |v:5_VVD| |draw:7_NN1|)
(|ncmod| _ |draw:7_NN1| |Sweden:6_NP1|)
(|ncmod| _ |Denmark:4_NP1| |media:2_NN|)
(|ncmod| _ |Denmark:4_NP1| |cheer:3_JJ|)
(|ncmod| _ |media:2_NN| |Euro-Scandinavian:1_JJ|)

("Denmark" "and" "Sweden" "tie" ".") 0 ; ()
(|conj| |and:2_CC| |Denmark:1_NP1|)
(|conj| |and:2_CC| |Sweden:3_NP1|)
```

Figure7-3: Grammatical relations of a Textual Entailment pair

The fact that one of the main principles of the proposed method is its reliance on the use of the grammatical relation between words, means that sentences that are not grammatically correct pose a problem. A domain highly represented by the challenge which often uses grammatically in-correct sentence structures is the news domain. An example of a text and hypothesis pair, along with their parsed representation can be seen in Figure 7-3. In this example, the system would fail to find the match between the two sentences despite its ability to recognise the similarity between the words ‘tie’ and ‘draw’. This is due, among other reasons, to the fact that the parser could not find any relation between the word ‘tie’ and one of the other words in the hypothesis sentence.

8. Limitations & Future work

As the evaluation section reveals, the system implemented to test the method, despite providing us with interesting information about the method, is far from perfect. There are several technological limitations which, if overcome, could provide us with better data about the method's validity.

One such limitation lies in the parsing process. Due to ill-structured sentences, spelling mistakes, lack of capitalisation and more the relationship between two words may be parsed in very different ways in similar sentences and sometimes the same word can be found to be of a certain part-of-speech in one sentence and another in a second sentence. Since the comparison is between grammatical relations and within parts-of-speech categories this will mean that the same grammatical relation may not match. This is demonstrated in Table 8-1 where the relation between 'visit' and 'China' is lost entirely in the example on the left due to the lack of capitalisation.

The best time to visit china.	The best time to visit China.
ncmod(china:N, time:N) obj(visit:V, time:N) xmod, to(time:N, visit:V) det(best:J, The:A)	ncmod(visit:V, time:N) **dobj(visit:V, China:N)** det(time:N, The:A) ncmod(time:N, best:J)

Table 8-1: Parsing error due to lack of capitalisation

This limitation highlights the methods high dependency on the use of proper grammatical rules which, given the un-edited and un-corrected state of the information published on the web, may be found to be a significant problem.

As explained in the background section, the neighbour data used for the set comparison is made up of related words rather than synonyms. This is not a vital problem as obviously two words which are close in meaning would have relatively similar neighbour sets. However, this does open the possibility of having two sentences saying the exact opposite returned as having similar meaning. A possible alternative to the neighbour data used, is the WordNet lexical database (WordNet 2006) which can be used to extract only words of a certain relationship to the original word. The decision regarding how important this

information is, will depend largely on the chosen application of the method. In the case of the question answering system this issue seemed to be of minor importance.

One thing that the neighbour data did provide and the system fails to take advantage of is the existence of phrases (i.e. cracked-up). Using phrases would make better use of redundancy as many words can be expressed as phrases and therefore these should be included in the word set comparison. Currently phrases can be associated as neighbours of a word, but they are not recognised as an element within a grammatical relation and therefore never expanded themselves.
In some cases the Rasp parser also provides the information to allow the use of phrases which could be used in the system by adding a specific rule to interpret it. As demonstrated in Table 8-2, when Rasp recognises a phrase it adds a relation describing it. This information could be used by the system to replace the verb 'crack+ed' in a given relation with the phrase '`crack+ed:V, up:R`'.

`Tom cracked up at hearing the joke.`	`Tom laughed at hearing the joke.`

`ncsubj(crack+ed:V, Tom:N)`
`xmod(crack+ed:V, at:I)`
`xcomp(at:I, hear+ing:V)`
`dobj(hear+ing:V, joke:N)`
`det(joke:N, the:A)`
`ncmod, prt(crack+ed:V, up:R)`
	`ncsubj(laugh+ed:V, Tom:N)`
	`xmod(laugh+ed:V, at:I)`
	`xcomp(at:I, hear+ing:V)`
	`dobj(hear+ing:V, joke:N)`
	`det(joke:N, the:A)`

Table 8-2: Grammatical Relations of a sentence with a phrase.

In addition to possible extensions to the method and the system which will be discussed later in this section, there are also other extensions which could be applied to improve the evaluation process. The focus of many such extensions lies in a limitation of the application of the system. It has been noted on several occasions in the evaluation section that running the system on a mass of data would provide for better evaluation of the method. The lack of such a test has been due to the large amount of resources (most significantly memory) required to perform it.
If the system was to be run on the web, a necessary extension would be the use of an additional system to extract only relevant information as noted in the analysis section. An alternative option is to run the system on the data collected in the British National Corpus. This might be a more viable solution to the problem as there is already a parsed

version of the data available. However, the nature of the collected data may mean that it would lack the redundancy required to take best advantage of the method.

There are many different levels of complexity possible to apply in the comparison between sentences, grammatical relations and individual words within the relations. Only a few of these were used in this system and incorporating other levels could be a basis for any future work. Among these possible levels of complexity are:

- **Proximity of word neighbours -** The Lexical comparison looks at the overlap of two sets of neighbour data. This is currently only made-up of those words directly linked to the original relation element. A possible extension would be to include the neighbours of each of the directly linked words in the set as well taking into account proximity to the original word.
- **Use of 'Wh' word information –** Any question answering system has to take into account the likely use of 'Wh' words at the start of the query sentence. Matching 'Wh' words with their possible categorical replacement (i.e. Who → person), or converting the grammatical structure of such sentences according to the 'Wh' word used are other possible extensions of this system.
- **Incorporating parsing confidence** - The Rasp parser can provide information about its confidence in the parsing of a sentence. This information could be incorporated into the calculation of the similarity between GRs.

9. Conclusion

The project has looked into the validity of using grammatical relations along with lexical similarity as a basis for a similarity measure between sentences. Its hypothesis that the natural redundancy existing within large collections of data could be utilised in a search for information has been confirmed to a small degree demonstrating that further study into it may prove to be fruitful.
In addition, the project has highlighted the strengths and weaknesses of the method which could be used in any future research into its use. It has pointed to precision as being the method's strongest point and exposed the reliance on correct grammatical structure as its weakness.

As the culmination of my degree, this project has been an application of the various theories and techniques I have studied in the last 3 years and has opened my eyes to how they can be applied in research. It has also taught me a great deal about the field of Natural Language Processing in which I see myself continuing, introducing me to techniques and resources used within the field and making me reflect on the direction I would like to pursue within it.

References & Bibliography

Chitu, A., (2006), 'Behind Google Q&A', In *Google Operating System* [Online] Available: http://googlesystem.blogspot.com/2006/10/behind-google-qa.html [20 April 2007]

Bird, S., and Loper, E., 'A Brief History of Natural Language Processing', In *The Language Challenge*, *System* [Online] Available: http://nltk.sourceforge.net/tutorial/introduction/section-x65.html [20 April 2007].

Briscoe, E. and Carroll, J. (2006) *RASP System Demonstration*, [Online], Available: http://www.informatics.susx.ac.uk/research/nlp/rasp/offline-demo.html [12 April 2007].

Briscoe, E., Carroll, J. and Watson, R. (2006) 'The Second Release of the RASP System', In *Proceedings of the COLING/ACL 2006 Interactive Presentation sessions*, Sydney, Australia.

Briscoe, E. (2006) *An introduction to tag sequence grammars and the RASP system parser*, University of Cambridge Computer laboratory Technical Report Number 662, [Online], Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-662.pdf [13 April 2007].

British Computer Society (BCS), (2006a) *BCS Code of Conduct*, [Online], Available: http://www.bcs.org/server.php?show=nav.6030 [12 April 2007].

British Computer Society (BCS), (2006b) *BCS Code of Good Practice*, [Online], Available: http://www.bcs.org/server.php?show=conWebDoc.1589 [12 April 2007].

Chambers Reference Online, (2006) Chambers Harrap Publishers Ltd, [Online], Available: http://www.chambersharrap.co.uk/chambers/index.shtml [12 April 2007].

Chomsky, Noam. (1957) *Syntactic Structures*. The Hague/Paris: Mouton.

Dagan, I., Glickman, O. and Magnini, B. (2006) 'The PASCAL Recognising Textual Entailment Challenge', *Lecture Notes in Computer Science*, vol 3944, January, pp 177 - 190.

Google, (2007a), [Online], Available: http://www.google.com/ [17 April 2007].

Google, (2007b), *Google Web Search Features: Q&A*, [Online], Available: http://www.google.com/help/features.html#qna [17 April 2007].

Hancox, P., (2006) A brief history of Natural Language Processing. [Online], Available: http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1_history.html (Up on 2/12/2006).

Hodges, A., (1995) *Alan Turing: a short biography*. [Online], Available: http://www.turing.org.uk/bio/ [20 April 2007].

Jurafsky, D. and Martin, J. (2000) *SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, New Jersey: Prentice-Hall.

Katz, B., Borchardt. G, Felshin, S., Et-Al, *Start - Natural Language Question Answering System,* InfoLab Group, MIT Computer Science and Artificial Intelligence Laboratory, [Online], Available: http://start.csail.mit.edu/ [17 April 2007].

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, J. and Temelkuran, B. (2002) 'Omnibase: Uniform Access to Heterogeneous Data for Question Answering', In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*, Stockholm, Sweden.

Kirste, B., (2003), *The Berlin Wall*, [Online], Available: http://userpage.chemie.fu-berlin.de/BIW/wall.html [20 April 2007].

Kwok, C., Etzioni, O., and Weld, D. (2006) *Scaling Question Answering to the Web,* University of Washington, [Online], Available: http://www.cs.washington.edu/homes/weld/papers/mulder-www10.pdf [17 April 2007].

Lin, D. (1998) 'Automatics Retrieval and Clustering of Similar Words' In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, Montreal, Quebec, Canada, pp. 768 - 774.

Manning C. and Schutze, H. (1999) *Foundations of statistical natural language processing*, Cambridge, MA: MIT Press.

Sharples, M., Hogg. D., Hutchinson, C., Torrance, S. and Young, D., (1996) *Computers and Thought: A Practical Introduction to Artificial Intelligence,* [Online], Available: http://www.informatics.susx.ac.uk/books/computers-and-thought/gloss/node1.html [19 April 2007].

The Official Google Blog, (2007), [Online], Available: http://googleblog.blogspot.com/ [20 April 2007].

Tomaiuolo, N., (2005), 'Google Unveils New Q&A Service', In *Information Today, Inc,* [Online], Available: http://newsbreaks.infotoday.com/nbReader.asp?ArticleId=16228 [20 April 2007]

Turing, A. M. (1950) 'Computing Machinery and Intelligence', *Mind*, vol. 59, October, pp. 433-460.

University Centre for Computer Corpus Research on Language (UCREL), (1993-2007) *UCREL CLAWS2 Tagset*, [Online], Available: http://www.comp.lancs.ac.uk/computing/research/ucrel/claws2tags.html [13 April 2007].

Weizenbaum, J. (1966) 'A Computer Program for the Study of Natural Language Communication Between Man and Machine', *Communications of the ACM*, vol 9, issue 1, January, pp. 36-45.

Winograd, T. (1972) *Understanding Natural Language*, Academic Press.

WordNet a lexical database for the English language, (2006), Princeton University, [Online], Available: http://wordnet.princeton.edu/ [19 April 2007]

Appendices

Appendix I – Experiment Results

1.1 Subjective Tests

Measure \ Question	When did the first man land on the moon ?		Who was the first man to land on the moon ?		What did Kennedy say to Webb ?	
Potential Correct answers	4		3		1	
	Sets	**Words**	**Sets**	**Words**	**Sets**	**Words**
Number of results returned	19	245	39	928	6	118
Index of top relevant result	12	1	2	5	1	1
Relevant results in top 20	2	2	1	1	1	1
Precision *	10.53%	10.00%	5.00%	5.00%	16.67%	5.00%
Recall	50.00%	50.00%	33.33%	33.33%	100.00%	100.00%
F-measure	17.39%	16.67%	8.70%	8.70%	28.57%	9.52%
Relevant results in top 5	0	1	1	1	1	1
Precision *	0.00%	5.00%	5.00%	5.00%	16.67%	5.00%
Recall	0.00%	25.00%	33.33%	33.33%	100.00%	100.00%
F-measure	0.00%	8.33%	8.70%	8.70%	28.57%	9.52%

Measure \ Question	Was the Apollo 11 moon landing faked by NASA ?		Was the landing on the moon faked by NASA ?		Was the Apollo 11 moon landing counterfeited by NASA ?	
Potential Correct answers	5		5		5	
	Sets	**Words**	**Sets**	**Words**	**Sets**	**Words**
Number of results returned	15	912	24	888	15	912
Index of top relevant result	1	3	15	3	1	5
Relevant results in top 20	2	3	2	3	2	2
Precision *	13.33%	15.00%	10.00%	15.00%	13.33%	10.00%
Recall	40.00%	60.00%	40.00%	60.00%	40.00%	40.00%
F-measure	20.00%	24.00%	16.00%	24.00%	20.00%	16.00%
Relevant results in top 5	1	1	0	3	1	1
Precision *	6.67%	5.00%	0.00%	15.00%	6.67%	5.00%
Recall	20.00%	20.00%	0.00%	60.00%	20.00%	20.00%
F-measure	10.00%	8.00%	0.00%	24.00%	10.00%	8.00%

Measure \ Question	Where did Woodstock festival take place ?		How many people died in Woodstock ?		Did any people die in Woodstock ?	
Potential Correct answers	3		2		2	
	Sets	**Words**	**Sets**	**Words**	**Sets**	**Words**
Number of results returned	0	249	5	194	9	208
Index of top relevant result	N/A	10	N/A	5	1	4
Relevant results in top 20	0	1	0	1	1	2
Precision *	0.00%	5.00%	0.00%	5.00%	11.11%	10.00%
Recall	0.00%	33.33%	0.00%	50.00%	50.00%	100.00%
F-measure	0.00%	8.70%	0.00%	9.09%	18.18%	18.18%
Relevant results in top 5	0	0	0	1	1	2
Precision *	0.00%	0.00%	0.00%	5.00%	11.11%	10.00%
Recall	0.00%	0.00%	0.00%	50.00%	50.00%	100.00%
F-measure	0.00%	0.00%	0.00%	9.09%	18.18%	18.18%

Measure \ Question	Did any one die in Woodstock ?		Were any people killed in Woodstock ?		Who played in Woodstock ?	
Potential Correct answers	2		2		3	
	Sets	**Words**	**Sets**	**Words**	**Sets**	**Words**
Number of results returned	4	169	4	873	5	155
Index of top relevant result	N/A	5	N/A	N/A	N/A	N/A
Relevant results in top 20	0	2	0	0	0	0
Precision *	0.00%	10.00%	0.00%	0.00%	0.00%	0.00%
Recall	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
F-measure	0.00%	18.18%	0.00%	0.00%	0.00%	0.00%
Relevant results in top 5	0	1	0	0	0	0
Precision *	0.00%	5.00%	0.00%	0.00%	0.00%	0.00%
Recall	0.00%	50.00%	0.00%	0.00%	0.00%	0.00%
F-measure	0.00%	9.09%	0.00%	0.00%	0.00%	0.00%

Question / Measure	What has Chancellor Helmut Kohl called the decision to open the Wall ?		When did the Berlin Wall fall ?		How many people died trying to cross the Berlin Wall ?	
Potential Correct answers	1		3		2	
	Sets	**Words**	**Sets**	**Words**	**Sets**	**Words**
Number of results returned	3	409	11	274	15	326
Index of top relevant result	1	2	7	8	3	19
Relevant results in top 20	1	1	1	1	1	1
Precision *	33.33%	5.00%	9.09%	5.00%	6.67%	5.00%
Recall	100.00%	100.00%	33.33%	33.33%	50.00%	50.00%
F-measure	50.00%	9.52%	14.29%	8.70%	11.76%	9.09%
Relevant results in top 5	1	1	0	0	1	0
Precision *	33.33%	5.00%	0.00%	0.00%	6.67%	0.00%
Recall	100.00%	100.00%	0.00%	0.00%	50.00%	0.00%
F-measure	50.00%	9.52%	0.00%	0.00%	11.76%	0.00%

Table 1: Subjective questions posed to the two comparison methods and their results (pages 62-64)

	Sets	Words
Average number of results	11.60	457.33
Number of relevant results in first place	33.33%	13.33%
Average index of top relevant result	4.40	5.46
Average relevant results in top 20	0.93	1.40
Precision	8.60%	7.00%
Recall	35.78%	54.00%
F-measure	13.87%	12.39%
Average relevant results in top 5	0.47	0.87
Precision	5.74%	4.33%
Recall	24.89%	37.22%
F-measure	9.33%	7.76%

Table 2: Summary of subjective test results

	Similarity score sum	Overlap count
Average number of results	11.60	22.13
Number of relevant results in first place	33.33%	33.33%
Average index of top relevant result	4.40	4.70
Average relevant results in top 20	0.93	1.00
Precision	8.60%	5.19%
Recall	35.78%	38.00%
F-measure	13.87%	9.13%

Table 3 Summary of proposed method test results using overlap count vs. similarity score summation

1.2 Objective Tests

The participants were asked to formulate 5 questions they would expect to be answered given texts about a given number of subjects.

Their questions were then passed through both the set comparison and the word count comparison methods.

Without knowing which method produced the answers, the participants were then asked to grade the answers on a score of 1-5, 1 being a bad answer and 5 a perfect one. They were not given guidance as to what constituted a good or bad answer.

A few notes:

* A relevant result was defined as one given a score of 5 (Perfect) by the participants.

* Recall could not be calculated as this would involve a great deal of work for each participant to go through all the texts looking for answers to their questions

Participant 1

Measure \ Question	Where was the woodstock festival first held at?		What was the name of the mission?		Which sides of germany was the berlin wall separating?	
	Sets	Words	Sets	Words	Sets	Words
Number of results returned	14	20	16	20	20	20
Index of top relevant result	1	6	1	N/A	11	N/A
Relevant results in top 20	1	1	2	0	1	0
Precision	7%	5%	13%	0%	5%	0%
Relevant results in top 5	1	0	2	0	0	0
Precision	7%	0%	13%	0%	0%	0%

Measure \ Question	How many years did the french revolution take?		In which year did Ghana celebrate its independence?	
	Sets	Words	Sets	Words
Number of results returned	20	20	7	20
Index of top relevant result	N/A	18	1	1
Relevant results in top 20	0	1	1	2
Precision	0%	5%	14%	10%
Relevant results in top 5	0	0	1	2
Precision	0%	0%	14%	10%

Participant 2

Measure \ Question	Who did Ghana get independence from?	
	Sets	Words
Number of results returned	1	20
Index of top relevant result	N/A	2
Relevant results in top 20	0	1
Precision	0%	5%
Relevant results in top 5	0	1
Precision	0%	5%

How many Woodstock festivals have there been?	
Sets	Words
20	20
N/A	N/A
0	0
0%	0%
0	0
0%	0%

Where is Woodstock ?	
Sets	Words
4	20
N/A	N/A
0	0
0%	0%
0	0
0%	0%

Measure \ Question	When did Ghana get independence?	
	Sets	Words
Number of results returned	3	20
Index of top relevant result	N/A	1
Relevant results in top 20	0	2
Precision	0%	10%
Relevant results in top 5	0	2
Precision	0%	10%

Who was King of France at the time of the French Revolution?	
Sets	Words
20	20
N/A	7
0	2
0%	10%
0	0
0%	0%

Participant 3

Measure \ Question	Who played the National anthem at Woodstock?	
	Sets	Words
Number of results returned	12	20
Index of top relevant result	N/A	N/A
Relevant results in top 20	0	0
Precision	0%	0%
Relevant results in top 5	0	0
Precision	0%	0%

What is the distance between the earth and the moon?	
Sets	Words
20	20
N/A	0
0	0
0%	0%
0	0
0%	0%

How many people went to Woodstock?	
Sets	Words
20	20
N/A	N/A
0	0
0%	0%
0	0
0%	0%

Measure \ Question	How long does it take to get from the earth to the moon?	
	Sets	Words
Number of results returned	17	20
Index of top relevant result	N/A	N/A
Relevant results in top 20	0	0
Precision	0%	0%
Relevant results in top 5	0	0
Precision	0%	0%

How long did Woodstock go on for?	
Sets	Words
17	20
N/A	N/A
0	0
0%	0%
0	0
0%	0%

Participant 4

Measure \ Question	Who played in Woodstock festival?		What is the name of the spacecraft that first time landed on the moon?		Why the Berlin wall fall?	
	Sets	**Words**	**Sets**	**Words**	**Sets**	**Words**
Number of results returned	12	20	20	20	0	20
Index of top relevant result	N/A	N/A	6	2	N/A	15
Relevant results in top 20	0	0	1	3	0	1
Precision	0%	0%	5%	15%	N/A	5%
Relevant results in top 5	0	0	0	2	0	0
Precision	0%	0%	0%	10%	N/A	0%

Measure \ Question	When did French revolution start?		Where is Ghana?	
	Sets	**Words**	**Sets**	**Words**
Number of results returned	0	20	4	20
Index of top relevant result	N/A	3	N/A	N/A
Relevant results in top 20	0	2	0	0
Precision	N/A	10%	0%	0%
Relevant results in top 5	0	2	0	0
Precision	N/A	10%	0%	0%

Table 4: Objective questions posed by participants to the two comparison methods and their results (pages 66-68)

	Sets	**Words**
Average number of results	12.35	20.00
Number of relevant results in first place	15.00%	10.00%
Average index of top relevant result	4.00	5.50
Average relevant results in top 20	0.30	0.75
Precision	2.20%	3.75%
Average relevant results in top 5	0.20	0.45
Precision	1.70%	2.25%

Table 5: Summary of objective test results

	Sets	**Words**
Bad	83.40%	83.50%
Not good	7.69%	5.50%
Medium	3.64%	5.25%
Good	2.83%	2.00%
Perfect	2.43%	3.75%

Table 6: Summary of grading by objective participants

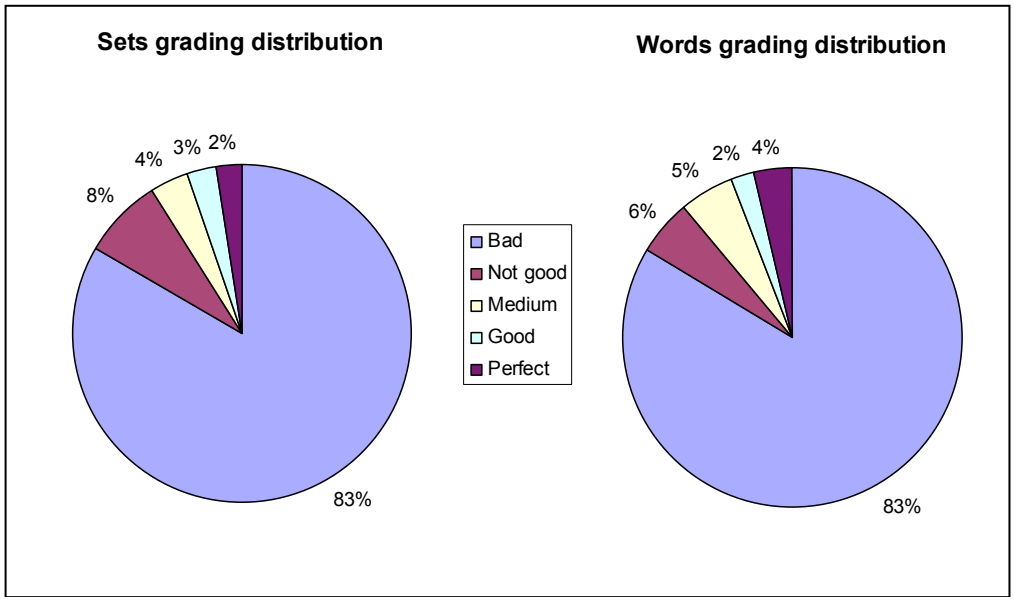

Figure 2: Objective tests grading distribution

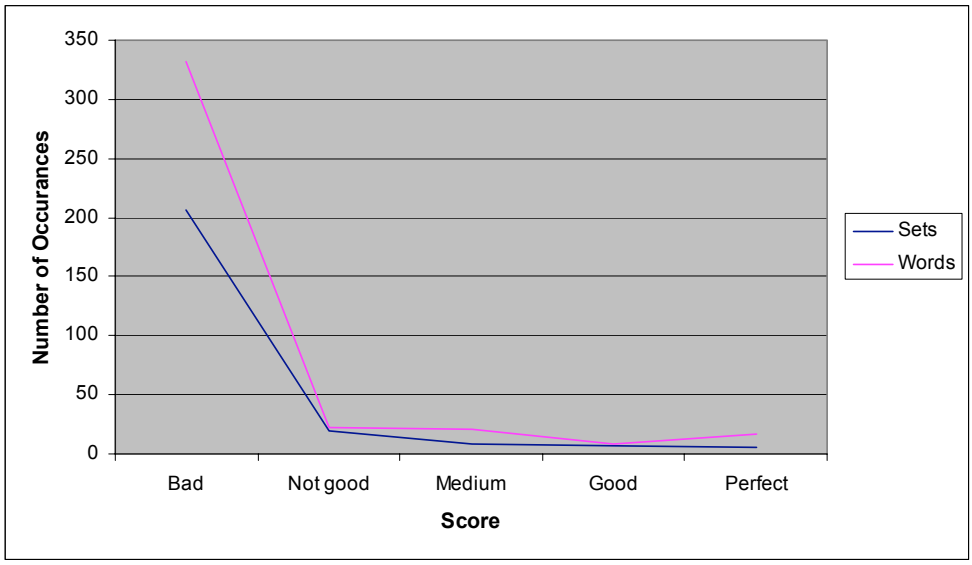

Figure 3: Comparative distribution of grades over all objective test results

Figure 4: Comparative distribution of higher grades over objective test results

1.3 Textual Entailment

The Pascal textual entailment challenge looks to see whether a system can successfully determine whether a hypothesis can be said to be entailed by some text. As such, the data provided is made up of pairs of text and hypothesis with a tag to indicate whether the hypothesis can be entailed by the text.

The system was run on the pairs and the score was stored and compared to the tag given in the data. A score of 0 was considered to be false (the hypothesis cannot be entailed from the text) and a score greater than 0 was considered to be true.

Correct[15]	50.90%
Wrong	49.10%

* Note: Pairs with more than 1 sentence in the text were removed from the test data.

Sample Data

```
<pair id="8" value="FALSE" task="IR">
        <t>Crude oil for April delivery traded at $37.80 a barrel, down 28 cents</t>
        <h>Crude oil prices rose to $37.80 per barrel</h>
</pair>
<pair id="13" value="TRUE" task="IR">
        <t>iTunes software has seen strong sales in Europe.</t>
        <h>Strong sales for iTunes in Europe.</h>
</pair>
<pair id="15" value="FALSE" task="IR">
        <t>All genetically modified food, including soya or maize oil produced from GM soya and maize, and food ingredients, must be labelled.</t>
        <h>Companies selling genetically modified foods don&amp;apos;t need labels.</h>
</pair>
<pair id="56" value="TRUE" task="IR">
        <t>Euro-Scandinavian media cheer Denmark v Sweden draw.</t>
        <h>Denmark and Sweden tie.</h>
</pair>
```

Sample Data Results

id = 8	value = FALSE	score = 0	➔	TRUE
id = 13	value = TRUE	score = 1.4	➔	TRUE
id = 15	value = FALSE	score = 0.25	➔	FALSE
id = 56	value = TRUE	score = 0	➔	FALSE

[15] Correct results are either: value = FALSE, score = 0 OR value = TRUE, score > 0

Appendix II – Sample Output [16]

Query: What has Chancellor Helmut Kohl called the decision to open the Wall ?

```
[obj]([call, V], [what, D]) (0.0)
[ncsubj]([call, V], [kohl, N]) (0.0)
[ncsubj]([open, V], [decision, N]) (0.0)
[xcomp, to]([call, V], [open, V]) (0.0)
[dobj]([call, V], [decision, N]) (0.0)
[dobj]([open, V], [wall, N]) (0.0)
[ncmod]([kohl, N], [chancellor, N]) (0.0)
[ncmod]([kohl, N], [helmut, N]) (0.0)

There are 3 matches. The best matches in order are:

1. Score: 4.0  Text: Berlin_Wall-bbc.txt     Sentence 11:
West German Chancellor Helmut Kohl has hailed the decision to open the Wall as \ historic
\ and called for a meeting with East German leader , Egon Krenz .
[ncmod]([krenz, N], [egon, N]) (0.0)
[ncsubj]([and, C], [kohl, N]) (0.0)
[conj]([and, C], [hail, V]) (0.0)
[conj]([and, C], [call, V]) (0.0)
[iobj]([call, V], [with, I]) (0.0)
[iobj]([call, V], [for, I]) (0.0)
[dobj]([with, I], [leader, N]) (0.0)
[ncmod]([leader, N], [east, N]) (0.0)
[ncmod]([leader, N], [german, J]) (0.0)
[dobj]([for, I], [meeting, N]) (0.0)
[ncsubj]([open, V], [decision, N]) (11.437640185967432)
[xcomp, to]([hail, V], [open, V]) (0.0)
[dobj]([hail, V], [decision, N]) (0.0)
[xcomp]([open, V], [as, C]) (0.0)
[dobj]([open, V], [wall, N]) (10.987045827564222)
[xcomp]([as, C], [historic, J]) (0.0)
[quote]([historic, J]) (0.0)
[ncmod]([kohl, N], [german, J]) (0.0)
[ncmod]([kohl, N], [chancellor, N]) (7.055720186985958)
[ncmod]([kohl, N], [helmut, N]) (4.0)
[ncmod]([german, J], [west, N]) (0.0)

2. Score: 0.5  Text: Berlin_Wall-about.txt   Sentence 9:
At an international press conference on June 15 , 1961 , Walter Ulbricht ( the leader of
the east German communist party , SED , and President of the Privy Council ) answered to
the question of a journalist : I understand your question as follows : there are people
in West Germany who want us to mobilize the construction workers of the GDR to build a
wall .
[ncmod]([answer, V], [at, I]) (0.0)
[ncsubj]([answer, V], [ulbricht, N]) (0.0)
[ncmod]([answer, V], [to, I]) (0.0)
[dobj]([to, I], [question, N]) (0.0)
[iobj]([question, N], [of, I]) (0.0)
[dobj]([of, I], [journalist, N]) (0.0)
[ncsubj]([understand, V], [i, P]) (0.0)
[ccomp]([understand, V], [follow, V]) (0.0)
[ncsubj]([follow, V], [question, N]) (0.0)
[comp]([follow, V], [be, V]) (0.0)
[ncsubj]([be, V], [there, EX]) (0.0)
[xcomp]([be, V], [people, N]) (0.0)
[cmod]([people, N], [in, I]) (0.0)
[ccomp]([in, I], [germany, N]) (0.0)
[ncsubj]([mobilize, V], [germany, N]) (0.0)
[ncsubj]([build, V], [worker, N]) (0.0)
[xcomp, to]([mobilize, V], [build, V]) (0.0)
[dobj]([mobilize, V], [worker, N]) (0.0)
[dobj]([build, V], [wall, N]) (4.299867398625968)
[ncmod]([worker, N], [construction, N]) (0.0)
[iobj]([worker, N], [of, I]) (0.0)
[dobj]([of, I], [gdr, N]) (0.0)
[cmod]([germany, N], [want, V]) (0.0)
[ncsubj]([want, V], [who, P]) (0.0)
[dobj]([want, V], [we, P]) (0.0)
[ncmod]([germany, N], [west, N]) (0.0)
[ncmod]([question, N], [as, C]) (0.0)
```

[16] Trimmed to up-to 5 answers per-question

```
[ncmod]([council, N], [and, C]) (0.0)
[conj]([and, C], [leader, N]) (0.0)
[conj]([and, C], [president, N]) (0.0)
[iobj]([president, N], [of, I]) (0.0)
[dobj]([of, I], [privy, N]) (0.0)
[iobj]([leader, N], [of, I]) (0.0)
[dobj]([of, I], [party, N]) (0.0)
[ncmod]([party, N], [german, J]) (0.0)
[ncmod]([party, N], [communist, J]) (0.0)
[ncmod]([german, J], [east, N]) (0.0)
[ncmod]([ulbricht, N], [walter, N]) (0.0)
[dobj]([at, I], [conference, N]) (0.0)
[ncmod]([conference, N], [on, I]) (0.0)
[dobj]([on, I], [june, N]) (0.0)
[ncmod, num]([june, N], [1961, M]) (0.0)
[ncmod, num]([june, N], [15, M]) (0.0)
[ncmod]([conference, N], [international, J]) (0.0)
[ncmod]([conference, N], [press, N]) (0.0)

3. Score: 0.5  Text: Berlin_Wall-wikipedia.txt      Sentence 0:
Berlin Wall From Wikipedia , the free encyclopedia ( Redirected from Berlin wall ) Jump
to : navigation , search East German construction workers building the Berlin Wall , 20
November 1961 .
[ncmod]([search, V], [november, N]) (0.0)
[ncmod, num]([november, N], [1961, M]) (0.0)
[ncmod, num]([november, N], [20, M]) (0.0)
[ncsubj]([search, V], [wall, N]) (0.0)
[ncsubj]([build, V], [worker, N]) (0.0)
[xcomp]([search, V], [build, V]) (0.0)
[dobj]([search, V], [worker, N]) (0.0)
[dobj]([build, V], [wall, N]) (4.299867398625968)
[ncmod]([wall, N], [berlin, N]) (0.0)
[ncmod]([worker, N], [german, J]) (0.0)
[ncmod]([worker, N], [construction, N]) (0.0)
[ncmod]([german, J], [east, N]) (0.0)
[ncmod]([wall, N], [from, I]) (0.0)
[dobj]([from, I], [wikipedia, N]) (0.0)
[ncsubj]([jump, V], [encyclopedia, N]) (0.0)
[ncmod, prt]([jump, V], [to, I]) (0.0)
[ncmod]([redirected, N], [from, I]) (0.0)
[dobj]([from, I], [wall, N]) (0.0)
[ncmod]([wall, N], [berlin, N]) (0.0)
[ncmod]([encyclopedia, N], [free, J]) (0.0)
[ncmod]([wall, N], [berlin, N]) (0.0)
```

Query: When did the Berlin Wall fall ?

```
[arg_mod]([fall, V], [when, R]) (0.0)
[ncsubj]([fall, V], [wall, N]) (0.0)
[ncmod]([wall, N], [berlin, N]) (0.0)

There are 11 matches. The best matches in order are:

1. Score: 6.0  Text: Berlin_Wall-wikipedia.txt      Sentence 0:
Berlin Wall From Wikipedia , the free encyclopedia ( Redirected from Berlin wall ) Jump
to : navigation , search East German construction workers building the Berlin Wall , 20
November 1961 .
[ncmod]([search, V], [november, N]) (0.0)
[ncmod, num]([november, N], [1961, M]) (0.0)
[ncmod, num]([november, N], [20, M]) (0.0)
[ncsubj]([search, V], [wall, N]) (0.0)
[ncsubj]([build, V], [worker, N]) (0.0)
[xcomp]([search, V], [build, V]) (0.0)
[dobj]([search, V], [worker, N]) (0.0)
[dobj]([build, V], [wall, N]) (0.0)
[ncmod]([wall, N], [berlin, N]) (6.646480863325991)
[ncmod]([worker, N], [german, J]) (0.0)
[ncmod]([worker, N], [construction, N]) (0.0)
[ncmod]([german, J], [east, N]) (0.0)
[ncmod]([wall, N], [from, I]) (0.0)
[dobj]([from, I], [wikipedia, N]) (0.0)
[ncsubj]([jump, V], [encyclopedia, N]) (0.0)
[ncmod, prt]([jump, V], [to, I]) (0.0)
[ncmod]([redirected, N], [from, I]) (0.0)
[dobj]([from, I], [wall, N]) (0.0)
[ncmod]([wall, N], [berlin, N]) (6.646480863325991)
[ncmod]([encyclopedia, N], [free, J]) (0.0)
```

```
[ncmod]([wall, N], [berlin, N]) (6.646480863325991)


2. Score: 2.6666666666666665  Text: French_Revolution-about.txt     Sentence 4:
A few histories stop in 1795 with the creation of the Directory , some stop in 1799 with
the creation of the Consulate , while many more stop in 1802 when Napoleon Bonaparte
became Consul for Life or 1804 when he became Emperor .
[arg_mod]([become, V], [when, R]) (4.510153230300875)
[ncsubj]([become, V], [he, P]) (0.0)
[dobj]([become, V], [emperor, N]) (0.0)
[cmod]([stop, N], [become, V]) (0.0)
[arg_mod]([become, V], [when, R]) (4.510153230300875)
[ncsubj]([become, V], [bonaparte, N]) (0.0)
[dobj]([become, V], [or, C]) (0.0)
[conj]([or, C], [consul, N]) (0.0)
[conj]([or, C], [1804, M]) (0.0)
[ncmod]([consul, N], [for, I]) (0.0)
[dobj]([for, I], [life, N]) (0.0)
[ncmod]([bonaparte, N], [napoleon, N]) (0.0)
[ncmod]([stop, N], [in, I]) (0.0)
[dobj]([in, I], [1802, M]) (0.0)
[ncmod]([stop, N], [many, D]) (0.0)
[ncmod]([stop, N], [more, D]) (0.0)
[ccomp]([few, D], [stop, V]) (0.0)
[ncsubj]([stop, V], [history, N]) (0.0)
[iobj]([stop, V], [with, I]) (0.0)
[iobj]([stop, V], [in, I]) (0.0)
[dobj]([with, I], [creation, N]) (0.0)
[ccomp]([creation, N], [stop, V]) (0.0)
[ncmod]([stop, V], [of, I]) (0.0)
[ncsubj]([stop, V], [some, D]) (0.0)
[ncmod]([stop, V], [with, I]) (0.0)
[dobj]([with, I], [creation, N]) (0.0)
[iobj]([creation, N], [of, I]) (0.0)
[dobj]([of, I], [consulate, N]) (0.0)
[ncmod]([stop, V], [in, I]) (0.0)
[dobj]([in, I], [1799, M]) (0.0)
[dobj]([of, I], [directory, N]) (0.0)
[dobj]([in, I], [1795, M]) (0.0)

3. Score: 2.0  Text: Battle_of_Hastings-bbc.txt     Sentence 39:
As the day drags on , the numbers began to tell and the English shield wall begins to
crack. * Late in the day , Harold takes an arrow in the eye and as his men mill around
him , four Norman knights break through and hack him down .
[ncsubj]([take, V], [harold, N]) (0.0)
[ccomp]([take, V], [and, C]) (0.0)
[dobj]([take, V], [arrow, N]) (0.0)
[ncmod]([and, C], [and, C]) (0.0)
[ncmod]([and, C], [around, I]) (0.0)
[ncsubj]([and, C], [knight, N]) (0.0)
[conj]([and, C], [break, V]) (0.0)
[conj]([and, C], [hack, V]) (0.0)
[ncmod]([hack, V], [down, R]) (0.0)
[dobj]([hack, V], [he, P]) (0.0)
[ncmod, prt]([break, V], [through, R]) (0.0)
[ncmod, num]([knight, N], [four, M]) (0.0)
[ncmod]([knight, N], [norman, J]) (0.0)
[dobj]([around, I], [he, P]) (0.0)
[conj]([and, C], [in, I]) (0.0)
[conj]([and, C], [as, C]) (0.0)
[dobj]([as, C], [mill, N]) (0.0)
[ncmod]([mill, N], [man, N]) (0.0)
[dobj]([in, I], [eye, N]) (0.0)
[conj]([and, C], [begin, V]) (0.0)
[conj]([and, C], [begin, V]) (0.0)
[ncsubj]([begin, V], [wall, N]) (6.606626553026477)
[xcomp, to]([begin, V], [crack., V]) (0.0)
[ncmod]([crack., V], [in, I]) (0.0)
[dobj]([in, I], [day, N]) (0.0)
[xcomp]([crack., V], [late, J]) (0.0)
[ncmod]([late, J], [*, R]) (0.0)
[ncmod]([wall, N], [english, J]) (0.0)
[ncmod]([wall, N], [shield, N]) (0.0)
[cmod]([begin, V], [as, C]) (0.0)
[ncsubj]([begin, V], [number, N]) (0.0)
[xcomp, to]([begin, V], [tell, V]) (0.0)
[ccomp]([as, C], [drag, V]) (0.0)
```

```
[ncsubj]([drag, V], [day, N]) (0.0)
[ncmod, prt]([drag, V], [on, R]) (0.0)

4. Score: 2.0  Text: Berlin_Wall-about.txt   Sentence 33:
At least 100 people were killed at the Berlin Wall , the last of them was Chris Gueffroy
( 1989-02-06 ) .
[cmod]([be, V], [at, I]) (0.0)
[ncsubj]([be, V], [last, M]) (0.0)
[xcomp]([be, V], [gueffroy, N]) (0.0)
[ncmod]([gueffroy, N], [chris, N]) (0.0)
[iobj]([last, M], [of, I]) (0.0)
[dobj]([of, I], [they, P]) (0.0)
[ccomp]([at, I], [kill, V]) (0.0)
[ncmod]([kill, V], [least, R]) (0.0)
[ncsubj]([kill, V], [people, N]) (0.0)
[iobj]([kill, V], [at, I]) (0.0)
[dobj]([at, I], [wall, N]) (0.0)
[ncmod]([wall, N], [berlin, N]) (6.646480863325991)
[ncmod, num]([people, N], [100, M]) (0.0)

5. Score: 2.0  Text: Berlin_Wall-about.txt   Sentence 43:
On February 1997 , a red line was painted on the pavement at the former \ Checkpoint
Charlie \ to mark the course of the former Berlin wall .
[obj2]([mark, V], [wall, N]) (0.0)
[dobj]([mark, V], [course, N]) (0.0)
[ncmod]([wall, N], [berlin, N]) (6.646480863325991)
[ncmod]([course, N], [of, I]) (0.0)
[ncmod]([charlie, N], [checkpoint, N]) (0.0)
[ncsubj]([paint, V], [line, N]) (0.0)
[ncmod]([paint, V], [at, I]) (0.0)
[dobj]([at, I], [former, D]) (0.0)
[ncmod]([paint, V], [on, I]) (0.0)
[dobj]([on, I], [pavement, N]) (0.0)
[ncmod]([line, N], [red, J]) (0.0)
[ncmod]([february, N], [1997, M]) (0.0)
[ncmod]([february, N], [on, N]) (0.0)
```

Appendix III - Sample Text

1.1 Original Text

Source: (Kirste 2003)

1989: Berliners celebrate the fall of the Wall
The Berlin Wall has been breached after nearly three decades keeping East and West Berliners apart.

At midnight East Germany's Communist rulers gave permission for gates along the Wall to be opened after hundreds of people converged on crossing points.

They surged through cheering and shouting and were be met by jubilant West Berliners on the other side.

Ecstatic crowds immediately began to clamber on top of the Wall and hack large chunks out of the 28-mile (45-kilometre) barrier.

It had been erected in 1961 on the orders of East Germany's former leader Walter Ulbricht stop people leaving for West Germany.

Since 1949 about 2.5 million people had fled East Germany.

After 1961, the Wall and other fortifications along the 860-mile (1,380-kilometre) border shared by East and West Germany have kept most East Germans in.

Many of those attempting to escape have been shot dead by border guards.

Exodus

The first indication that change was imminent came earlier today when East Berlin's Communist party spokesman, Gunther Schabowski, announced East Germans would be allowed to travel directly to West Germany.

The move was intended to stem an exodus into West Germany through the "back door" which began last summer when the new and more liberal regime in Hungary opened its border.

The flow of migrants was intensified last week when Czechoslovakia also granted free access to West Germany through its border.

West German Chancellor Helmut Kohl has hailed the decision to open the Wall as "historic" and called for a meeting with East German leader, Egon Krenz.

1.2 Parsed by RASP

```
(|1989:1 MC| |::2 :| |Berliner+s:3 NN2| |celebrate:4 VV0| |the:5 AT| |fall:6_NN1|
|of:7_IO| |the:8_AT| |Wall:9_NP1| |The:10_AT| |Berlin:11_NP1| |Wall:12_NP1|
|have+s:13_VHZ| |be+en:14_VBN| |breach+ed:15_VVN| |after:16_ICS| |nearly:17_RR|
|three:18 MC| |decade+s:19 NN2| |keep+ing:20 VVG| |East:21_ND1| |and:22_CC| |West:23_NP1|
|Berliner+s:24_NN2| |apart:25_RL| |.:26_.|) 1 ; (-56.604)

(|ncsubj| |breach+ed:15 VVN| |1989:1 MC|  )
(|ncmod| _ |breach+ed:15_VVN| |after:16_ICS|)
(|dobj| |after:16_ICS| |decade+s:19_NN2|)
(|ncmod|   |decade+s:19 NN2| |nearly:17 RR|)
(|ncmod| |num| |decade+s:19 NN2| |three:18 MC|)
(|ncsubj| |keep+ing:20 VVG| |decade+s:19 NN2| _)
(|xmod| _ |decade+s:19_NN2| |keep+ing:20_VVG|)
(|ncmod| |prt| |keep+ing:20_VVG| |apart:25_RL|)
(|obj2| |keep+ing:20 VVG| |Berliner+s:24_NN2|)
(|dobj| |keep+ing:20 VVG| |and:22 CC|)
(|conj| |and:22 CC| |East:21 ND1|)
(|conj| |and:22 CC| |West:23 NP1|)
(|aux| |breach+ed:15_VVN| |have+s:13_VHZ|)
(|aux| |breach+ed:15_VVN| |be+en:14_VBN|)
(|passive| |breach+ed:15 VVN|)
(|ta| |colon| |1989:1 MC| |celebrate:4 VV0|)
(|ncsubj| |celebrate:4 VV0| |Berliner+s:3_NN2| _)
(|obj2| |celebrate:4 VV0| |Wall:12 NP1|)
(|dobj| |celebrate:4_VV0| |fall:6_NN1|)
(|det| |Wall:12_NP1| |The:10_AT|)
(|ncmod|   |Wall:12 NP1| |Berlin:11_NP1|)
(|det| |fall:6 NN1| |the:5 AT|)
(|iobj| |fall:6 NN1| |of:7 IO|)
(|dobj| |of:7 IO| |Wall:9 NP1|)
(|det| |Wall:9_NP1| |the:8_AT|)

(|At:1 II| |midnight:2 NNT1| |East:3 NP1| |Germany:4 NP1| |'s+:5 $| |Communist:6_JJ|
|ruler+s:7 NN2| |give+ed:8 VVD| |permission:9 NN1| |for:10 IF| |gate+s:11 VVZ|
|along:12 II| |the:13 AT| |Wall:14 NP1| |to:15 TO| |be:16 VB0| |open+ed:17 VVN|
|after:18_ICS| |hundred+s:19_NNO2| |of:20_IO| |people:21_NN| |converge+ed:22_VVN|
|on:23_II| |cross+ing:24_VVG| |point+s:25_NN2| |.:26_.|) 1 ; (-44.139)

(|ncmod|   |give+ed:8 VVD| |At:1 II|)
(|ncsubj| |give+ed:8 VVD| |ruler+s:7 NN2|  )
(|dobj| |give+ed:8 VVD| |permission:9 NN1|)
(|ccomp| _ |permission:9_NN1| |gate+s:11_VVZ|)
(|ncsubj| |gate+s:11_VVZ| |for:10_IF| |inv|)
(|cmod|   |gate+s:11 VVZ| |along:12 II|)
(|ccomp|   |along:12 II| |Wall:14 NP1|)
(|ncsubj| |open+ed:17 VVN| |Wall:14 NP1|  )
(|ncmod|   |open+ed:17 VVN| |after:18 ICS|)
(|dobj| |after:18_ICS| |hundred+s:19_NNO2|)
(|ncmod| _ |hundred+s:19_NNO2| |of:20_IO|)
(|dobj| |of:20 IO| |people:21 NN|)
(|passive| |converge+ed:22 VVN|)
(|ncsubj| |converge+ed:22 VVN| |people:21 NN| |obj|)
(|xmod|   |people:21 NN| |converge+ed:22 VVN|)
(|xcomp| _ |converge+ed:22_VVN| |on:23_II|)
(|xcomp| _ |on:23_II| |cross+ing:24_VVG|)
(|dobj| |cross+ing:24 VVG| |point+s:25_NN2|)
(|aux| |open+ed:17 VVN| |be:16_VB0|)
(|passive| |open+ed:17 VVN|)
(|det| |Wall:14_NP1| |the:13_AT|)
(|ncmod| |poss| |ruler+s:7_NN2| |Germany:4_NP1|)
(|ncmod|   |ruler+s:7 NN2| |Communist:6 JJ|)
(|ncmod|   |Germany:4 NP1| |East:3 NP1|)
(|dobj| |At:1_II| |midnight:2_NNT1|)

(|They:1_PPHS2| |surge+ed:2_VVD| |through:3_II| |cheering:4_JJ| |and:5_CC|
|shout+ing:6_VVG| |and:7_CC| |be+ed:8_VBDR| |be:9_VB0| |meet+ed:10_VVN| |by:11_II|
|jubilant:12 JJ| |West:13 ND1| |Berliner+s:14_NN2| |on:15_II| |the:16_AT| |other:17_JB|
|side:18_NN1| |.:19_.|) 1 ; (-36.919)

(|conj| |and:5 CC| |surge+ed:2 VVD|)
(|conj| |and:5_CC| |meet+ed:10_VVN|)
(|xsubj| |meet+ed:10_VVN| |and:7_CC| _)
```

```
(|aux| |meet+ed:10_VVN| |be:9_VB0|)
(|passive| |meet+ed:10 VVN|)
(|iobj| |meet+ed:10 VVN| |on:15 II|)
(|iobj| |meet+ed:10 VVN| |by:11 II|)
(|dobj| |on:15 II| |side:18 NN1|)
(|det| |side:18_NN1| |the:16_AT|)
(|ncmod| _ |side:18_NN1| |other:17_JB|)
(|dobj| |by:11 II| |Berliner+s:14 NN2|)
(|ncmod|   |Berliner+s:14 NN2| |jubilant:12 JJ|)
(|ncmod|   |Berliner+s:14 NN2| |West:13_ND1|)
(|conj| |and:7_CC| |shout+ing:6_VVG|)
(|conj| |and:7_CC| |be+ed:8_VBDR|)
(|ncsubj| |surge+ed:2 VVD| |They:1 PPHS2|  )
(|ncmod|   |surge+ed:2 VVD| |through:3 II|)
(|dobj| |through:3_II| |cheering:4_JJ|)

(|Ecstatic:1_NP1| |crowd+s:2_NN2| |immediately:3_RR| |begin+ed:4_VVD| |to:5_TO|
|clamber:6_VV0| |on:7_II| |top:8_NN| |of:9_IO| |the:10_AT| |Wall:11_NP1| |and:12_CC|
|hack:13 VV0| |large:14 JJ| |chunk+s:15 NN2| |out:16 RP| |of:17 IO| |the:18 AT| |28-
mile:19_JB| |(:20_(| |45-kilometre:21_JB| |):22_)| |barrier:23_NN1| |.:24_.|) 1 ; (-
49.794)

(|ncsubj| |begin+ed:4_VVD| |crowd+s:2_NN2| _)
(|ncmod| _ |begin+ed:4_VVD| |immediately:3_RR|)
(|xcomp| |to| |begin+ed:4 VVD| |and:12_CC|)
(|conj| |and:12 CC| |clamber:6 VV0|)
(|conj| |and:12 CC| |hack:13 VV0|)
(|dobj| |hack:13 VV0| |chunk+s:15 NN2|)
(|ncmod| _ |chunk+s:15_NN2| |large:14_JJ|)
(|iobj| |chunk+s:15_NN2| |of:17_IO|)
(|ncmod|   |of:17 IO| |out:16 RP|)
(|dobj| |of:17 IO| |barrier:23 NN1|)
(|det| |barrier:23 NN1| |the:18 AT|)
(|ncmod|   |barrier:23 NN1| |28-mile:19 JB|)
(|ta| |bal| |28-mile:19_JB| |45-kilometre:21_JB|)
(|iobj| |clamber:6_VV0| |on:7_II|)
(|dobj| |on:7 II| |top:8 NN|)
(|iobj| |top:8 NN| |of:9 IO|)
(|dobj| |of:9 IO| |Wall:11 NP1|)
(|det| |Wall:11_NP1| |the:10_AT|)
(|ncmod| _ |crowd+s:2_NN2| |Ecstatic:1_NP1|)

(|It:1 PPH1| |have+ed:2 VHD| |be+en:3 VBN| |erect+ed:4 VVN| |in:5 II| |1961:6 MC|
|on:7 II| |the:8 AT| |order+s:9 NN2| |of:10 IO| |East:11 NP1| |Germany:12 NP1| |'s+:13_$|
|former:14 DA| |leader:15 NN1| |Walter:16 NP1| |Ulbricht:17 NP1| |stop:18 VV0|
|people:19_NN| |leave+ing:20_VVG| |for:21_IF| |West:22_NP1| |Germany:23_NP1| |.:24_.|) 1
; (-35.991)

(|ncsubj| |erect+ed:4 VVN| |It:1 PPH1| _)
(|ncmod|   |erect+ed:4 VVN| |on:7 II|)
(|dobj| |on:7 II| |order+s:9 NN2|)
(|det| |order+s:9_NN2| |the:8_AT|)
(|iobj| |order+s:9_NN2| |of:10_IO|)
(|dobj| |of:10 IO| |leader:15 NN1|)
(|ncmod| |poss| |leader:15 NN1| |Germany:12_NP1|)
(|ncmod|   |leader:15 NN1| |former:14 DA|)
(|ccomp|   |leader:15 NN1| |stop:18 VV0|)
(|ncsubj| |stop:18_VV0| |Ulbricht:17_NP1| _)
(|dobj| |stop:18_VV0| |people:19_NN|)
(|ncsubj| |leave+ing:20 VVG| |people:19 NN| _)
(|xmod|   |people:19 NN| |leave+ing:20 VVG|)
(|iobj| |leave+ing:20 VVG| |for:21 IF|)
(|dobj| |for:21_IF| |Germany:23_NP1|)
(|ncmod| _ |Germany:23_NP1| |West:22_NP1|)
(|ncmod|   |Ulbricht:17 NP1| |Walter:16 NP1|)
(|ncmod|   |Germany:12 NP1| |East:11 NP1|)
(|ncmod|   |erect+ed:4 VVN| |in:5_II|)
(|dobj| |in:5 II| |1961:6 MC|)
(|aux| |erect+ed:4_VVN| |have+ed:2_VHD|)
(|aux| |erect+ed:4_VVN| |be+en:3_VBN|)
(|passive| |erect+ed:4_VVN|)

(|Since:1 ICS| |1949:2 MC| |about:3 II| |2.5:4 MC| |million:5 NNO| |people:6 NN|
|have+ed:7_VHD| |flee+ed:8_VVN| |East:9_ND1| |Germany:10_NP1| |.:11_.|) 1 ; (-22.884)

(|ncmod| _ |flee+ed:8_VVN| |Since:1_ICS|)
(|ncsubj| |flee+ed:8_VVN| |about:3_II| |inv|)
```

```
(|aux| |flee+ed:8_VVN| |have+ed:7_VHD|)
(|obj| |flee+ed:8 VVN| |Germany:10 NP1|)
(|ncmod|   |Germany:10 NP1| |East:9_ND1|)
(|dobj| |about:3 II| |people:6 NN|)
(|ncmod| |num| |people:6 NN| |2.5:4 MC|)
(|ncmod| |num| |people:6_NN| |million:5_NNO|)
(|dobj| |Since:1_ICS| |1949:2_MC|)

(|After:1 ICS| |1961:2 MC| |,:3 ,| |the:4 AT| |Wall:5 NP1| |and:6 CC| |other:7 JB|
|fortification+s:8 NN2| |along:9 II| |the:10 AT| |860-mile:11 JB| |(:12 (| |1,380-
kilometre:13_JB| |):14_)| |border:15_NN1| |share+ed:16_VVN| |by:17_II| |East:18_ND1|
|and:19_CC| |West:20_NP1| |Germany:21_NP1| |have:22_VH0| |keep+ed:23_VVN| |most:24_RR|
|East:25_ND1| |German+s:26_NN2| |in:27_II| |.:28_.|) 1 ; (-68.675)

(|ncmod|   |keep+ed:23 VVN| |After:1 ICS|)
(|ncsubj| |keep+ed:23 VVN| |and:6 CC|  )
(|ncmod| |prt| |keep+ed:23_VVN| |in:27_II|)
(|aux| |keep+ed:23_VVN| |have:22_VH0|)
(|obj| |keep+ed:23 VVN| |German+s:26 NN2|)
(|ncmod|   |German+s:26 NN2| |most:24 RR|)
(|ncmod|   |German+s:26 NN2| |East:25_ND1|)
(|det| |and:6 CC| |the:4 AT|)
(|conj| |and:6_CC| |Wall:5_NP1|)
(|conj| |and:6_CC| |fortification+s:8_NN2|)
(|ncmod|   |fortification+s:8 NN2| |along:9_II|)
(|dobj| |along:9 II| |and:19 CC|)
(|det| |and:19 CC| |the:10 AT|)
(|ncmod|   |and:19 CC| |860-mile:11 JB|)
(|conj| |and:19_CC| |border:15_NN1|)
(|conj| |and:19_CC| |East:18_ND1|)
(|conj| |and:19 CC| |Germany:21 NP1|)
(|ncmod|   |Germany:21 NP1| |West:20_NP1|)
(|passive| |share+ed:16 VVN|)
(|ncsubj| |share+ed:16 VVN| |border:15 NN1| |obj|)
(|xmod| _ |border:15_NN1| |share+ed:16_VVN|)
(|ncmod| |prt| |share+ed:16_VVN| |by:17_II|)
(|ta| |bal| |860-mile:11 JB| |1,380-kilometre:13_JB|)
(|ncmod|   |fortification+s:8 NN2| |other:7_JB|)
(|dobj| |After:1_ICS| |1961:2_MC|)

(|Many:1_DA2| |of:2_IO| |those:3_DD2| |attempt+ing:4_VVG| |to:5_TO| |escape:6_VV0|
|have:7 VH0| |be+en:8 VBN| |shoot+ed:9 VVN| |dead:10_JJ| |by:11_II| |border:12_NN1|
|guard+s:13_NN2| |.:14_.|) 1 ; (-22.427)

(|ncsubj| |shoot+ed:9 VVN| |Many:1 DA2| _)
(|ncmod| _ |shoot+ed:9_VVN| |by:11_II|)
(|dobj| |by:11_II| |guard+s:13_NN2|)
(|ncmod|   |guard+s:13 NN2| |border:12_NN1|)
(|aux| |shoot+ed:9 VVN| |have:7 VH0|)
(|aux| |shoot+ed:9 VVN| |be+en:8_VBN|)
(|passive| |shoot+ed:9 VVN|)
(|xcomp| _ |shoot+ed:9_VVN| |dead:10_JJ|)
(|arg| _ |escape:6_VV0| |Many:1_DA2|)
(|xmod| |to| |Many:1 DA2| |escape:6_VV0|)
(|cmod|   |Many:1 DA2| |of:2 IO|)
(|ccomp|   |of:2 IO| |those:3 DD2|)
(|ncsubj| |attempt+ing:4_VVG| |those:3_DD2| _)

(|Exodus:1_NP1|) 0 ; ()


(|The:1 AT| |first:2 MD| |indication:3 NN1| |that:4 CST| |change:5 NN1| |be+ed:6 VBDZ|
|imminent:7_JJ| |come+ed:8_VVD| |earlier:9_RRR| |today:10_RT| |when:11_CS| |East:12_NP1|
|Berlin:13_NP1| |'s+:14_$| |Communist:15_JJ| |party:16_NN1| |spokesman:17_NN1| |,:18_,|
|Gunther:19 NP1| |Schabowski:20 NP1| |,:21 ,| |announce+ed:22 VVD| |East:23 NP1|
|German+s:24 NN2| |would:25 VM| |be:26 VB0| |allow+ed:27 VVN| |to:28 TO| |travel:29_VV0|
|directly:30_RR| |to:31_II| |West:32_NP1| |Germany:33_NP1| |.:34_.|) 1 ; (-55.637)

(|ncsubj| |come+ed:8_VVD| |indication:3_NN1| _)
(|cmod| _ |come+ed:8_VVD| |when:11_CS|)
(|ccomp|   |when:11 CS| |announce+ed:22 VVD|)
(|ncsubj| |announce+ed:22 VVD| |spokesman:17 NN1| _)
(|ccomp|   |announce+ed:22 VVD| |allow+ed:27 VVN|)
(|ncsubj| |allow+ed:27 VVN| |German+s:24_NN2| _)
(|aux| |allow+ed:27_VVN| |would:25_VM|)
(|aux| |allow+ed:27_VVN| |be:26_VB0|)
(|passive| |allow+ed:27_VVN|)
```

```
(|xcomp| |to| |allow+ed:27_VVN| |travel:29_VV0|)
(|ncmod|   |travel:29 VV0| |to:31 II|)
(|ncmod|   |to:31 II| |directly:30 RR|)
(|dobj| |to:31 II| |Germany:33 NP1|)
(|ncmod|   |Germany:33 NP1| |West:32 NP1|)
(|ncmod| _ |German+s:24_NN2| |East:23_NP1|)
(|ta| |bal| |spokesman:17_NN1| |Schabowski:20_NP1|)
(|ncmod|   |Schabowski:20 NP1| |Gunther:19 NP1|)
(|ncmod| |poss| |spokesman:17 NN1| |Berlin:13 NP1|)
(|ncmod|   |spokesman:17 NN1| |Communist:15 JJ|)
(|ncmod| _ |spokesman:17_NN1| |party:16_NN1|)
(|ncmod| _ |Berlin:13_NP1| |East:12_NP1|)
(|ncmod|   |come+ed:8 VVD| |today:10 RT|)
(|ncmod|   |come+ed:8 VVD| |earlier:9_RRR|)
(|det| |indication:3 NN1| |The:1 AT|)
(|ncmod| |num| |indication:3 NN1| |first:2 MD|)
(|ccomp| |that:4_CST| |indication:3_NN1| |be+ed:6_VBDZ|)
(|ncsubj| |be+ed:6_VBDZ| |change:5_NN1| _)
(|xcomp| _ |be+ed:6_VBDZ| |imminent:7_JJ|)

(|The:1 AT| |move:2 NN1| |be+ed:3 VBDZ| |intend+ed:4 VVN| |to:5 TO| |stem:6 VV0|
|an:7 AT1| |exodus:8 NN1| |into:9 II| |West:10 NP1| |Germany:11 NP1| |through:12_II|
|the:13_AT| |":14_"| |back:15_NN1| |door:16_NN1| |":17_"| |which:18_DDQ|
|begin+ed:19_VVD| |last:20_MD| |summer:21_NNT1| |when:22_RRQ| |the:23_AT| |new:24_JJ|
|and:25 CC| |more:26 DAR| |liberal:27 JJ| |regime:28 NN1| |in:29_II| |Hungary:30_NP1|
|open+ed:31_VVD| |its:32_APP$| |border:33_NN1| |.:34_.|) 0 ; ()

(|ncsubj| |begin+ed:19 VVD| |which:18 DDQ|  )
(|ncmod| _ |begin+ed:19_VVD| |summer:21_NNT1|)
(|cmod| _ |summer:21_NNT1| |open+ed:31_VVD|)
(|arg mod|   |open+ed:31 VVD| |when:22 RRQ|)
(|ncsubj| |open+ed:31 VVD| |regime:28 NN1| _)
(|dobj| |open+ed:31 VVD| |border:33 NN1|)
(|det| |border:33 NN1| |its:32 APP$|)
(|det| |regime:28_NN1| |the:23_AT|)
(|ncmod| _ |regime:28_NN1| |in:29_II|)
(|dobj| |in:29 II| |Hungary:30 NP1|)
(|ncmod|   |regime:28 NN1| |and:25_CC|)
(|conj| |and:25 CC| |new:24 JJ|)
(|conj| |and:25_CC| |liberal:27_JJ|)
(|ncmod| _ |liberal:27_JJ| |more:26_DAR|)
(|xcomp|   |begin+ed:19 VVD| |last:20 MD|)
(|ncmod|   |door:16 NN1| |back:15 NN1|)
(|ncsubj| |intend+ed:4 VVN| |move:2 NN1| _)
(|aux| |intend+ed:4 VVN| |be+ed:3_VBDZ|)
(|passive| |intend+ed:4_VVN|)
(|xcomp| |to| |intend+ed:4_VVN| |stem:6_VV0|)
(|ncmod| |prt| |stem:6 VV0| |through:12_II|)
(|iobj| |stem:6 VV0| |into:9 II|)
(|dobj| |stem:6 VV0| |exodus:8 NN1|)
(|dobj| |into:9 II| |Germany:11 NP1|)
(|ncmod| _ |Germany:11_NP1| |West:10_NP1|)
(|det| |exodus:8_NN1| |an:7_AT1|)
(|det| |move:2_NN1| |The:1_AT|)

(|The:1 AT| |flow:2 NN1| |of:3 IO| |migrant+s:4 NN2| |be+ed:5 VBDZ| |intensify+ed:6_VVN|
|last:7 MD| |week:8 NNT1| |when:9 RRQ| |Czechoslovakia:10 NP1| |also:11 RR|
|grant+ed:12_VVN| |free:13_JJ| |access:14_NN1| |to:15_II| |West:16_NP1| |Germany:17_NP1|
|through:18_II| |its:19_APP$| |border:20_NN1| |.:21_.|) 1 ; (-32.716)

(|ncsubj| |intensify+ed:6 VVN| |flow:2 NN1|  )
(|ncmod|   |intensify+ed:6 VVN| |week:8 NNT1|)
(|cmod| _ |week:8_NNT1| |grant+ed:12_VVN|)
(|arg_mod| _ |grant+ed:12_VVN| |when:9_RRQ|)
(|ncsubj| |grant+ed:12 VVN| |Czechoslovakia:10_NP1| _)
(|ncmod|   |grant+ed:12 VVN| |also:11 RR|)
(|iobj| |grant+ed:12 VVN| |through:18 II|)
(|dobj| |grant+ed:12 VVN| |access:14 NN1|)
(|dobj| |through:18_II| |border:20_NN1|)
(|det| |border:20_NN1| |its:19_APP$|)
(|ncmod|   |access:14 NN1| |to:15 II|)
(|dobj| |to:15 II| |Germany:17 NP1|)
(|ncmod|   |Germany:17 NP1| |West:16 NP1|)
(|ncmod|   |access:14 NN1| |free:13 JJ|)
(|ncmod| |num| |week:8_NNT1| |last:7_MD|)
(|aux| |intensify+ed:6_VVN| |be+ed:5_VBDZ|)
(|passive| |intensify+ed:6_VVN|)
```

```
(|det| |flow:2_NN1| |The:1_AT|)
(|iobj| |flow:2 NN1| |of:3 IO|)
(|dobj| |of:3_IO| |migrant+s:4_NN2|)

(|West:1 ND1| |German:2 JJ| |Chancellor:3 NNS1| |Helmut:4 NP1| |Kohl:5 NP1|
|have+s:6_VHZ| |hail+ed:7_VVN| |the:8_AT| |decision:9_NN1| |to:10_TO| |open:11_VV0|
|the:12_AT| |Wall:13_NP1| |as:14_CSA| |":15_"| |historic:16_JJ| |":17_"| |and:18_CC|
|call+ed:19 VVN| |for:20 IF| |a:21 AT1| |meeting:22 NN1| |with:23 IW| |East:24 NP1|
|German:25_JJ| |leader:26_NN1| |,:27_,| |Egon:28_NP1| |Krenz:29_NP1| |.:30_.|) 1 ; (-
52.283)

(|ta| |voc| |and:18_CC| |Krenz:29_NP1|)
(|ncmod|   |Krenz:29 NP1| |Egon:28 NP1|)
(|ncsubj| |and:18 CC| |Kohl:5 NP1| _)
(|aux| |and:18 CC| |have+s:6 VHZ|)
(|conj| |and:18 CC| |hail+ed:7 VVN|)
(|conj| |and:18_CC| |call+ed:19_VVN|)
(|iobj| |call+ed:19_VVN| |with:23_IW|)
(|iobj| |call+ed:19 VVN| |for:20 IF|)
(|dobj| |with:23 IW| |leader:26 NN1|)
(|ncmod|   |leader:26 NN1| |East:24 NP1|)
(|ncmod|   |leader:26 NN1| |German:25_JJ|)
(|dobj| |for:20_IF| |meeting:22_NN1|)
(|det| |meeting:22_NN1| |a:21_AT1|)
(|ncsubj| |open:11 VV0| |decision:9 NN1|  )
(|xcomp| |to| |hail+ed:7 VVN| |open:11 VV0|)
(|dobj| |hail+ed:7 VVN| |decision:9 NN1|)
(|xcomp|   |open:11 VV0| |as:14 CSA|)
(|dobj| |open:11_VV0| |Wall:13_NP1|)
(|xcomp| _ |as:14_CSA| |historic:16_JJ|)
(|quote| |historic:16 JJ|)
(|det| |Wall:13 NP1| |the:12 AT|)
(|det| |decision:9 NN1| |the:8 AT|)
(|ncmod|   |Kohl:5 NP1| |German:2 JJ|)
(|ncmod| _ |Kohl:5_NP1| |Chancellor:3_NNS1|)
(|ncmod| _ |Kohl:5_NP1| |Helmut:4_NP1|)
(|ncmod| _ |German:2_JJ| |West:1_ND1|)
```

Appendix IV - Sample Neighbours Data

```
fair = [thenceforth, R, 0.16107] [pleasurably, R, 0.15354]
        [anyways, R, 0.14276] [incognito, R, 0.12847] [astray, R, 0.11786]
        [selflessly, R, 0.11399] [impartially, R, 0.11166] [wetly, R, 0.11122]

lightly = [gently, R, 0.24528] [heavily, R, 0.17773] [again, R, 0.16818]
        [softly, R, 0.16279] [well, R, 0.15183] [carefully, R, 0.14199]
        [hard, R, 0.14026] [before, R, 0.13969]

quietly = [silently, R, 0.24206] [softly, R, 0.23717] [gently, R, 0.21400]
        [again, R, 0.19424] [slowly, R, 0.19295] [loudly, R, 0.18766]
        [quickly, R, 0.17108] [happily, R, 0.17004]

spontaneously = [shortly, R, 0.14899] [onwards, R, 0.14449]
        [furthermore, R, 0.14420] [speedily, R, 0.14252] [cautiously, R, 0.14132]
        [elsewhere, R, 0.14073] [informally, R, 0.13976] [moreover, R, 0.13527]

ceremonially = [ceremoniously, R, 0.18455] [asunder, R, 0.16414]
        [blearily, R, 0.16167] [pinafore, R, 0.15807] [cleanly, R, 0.13197] [irreverently,
        R, 0.11564] [shabbily, R, 0.11340] [hygienically, R, 0.10533]

furiously = [angrily, R, 0.29808] [hard, R, 0.24553] [again, R, 0.22022]
        [impatiently, R, 0.21539] [frantically, R, 0.20826] [fiercely, R, 0.20181]
        [vigorously, R, 0.19971] [straight, R, 0.18215]

mightily = [thither, R, 0.11698] [heartily, R, 0.08967] [hither, R, 0.08843]
        [tremendously, R, 0.08747] [immensely, R, 0.08656] [mortally, R, 0.08616]
        [greatly, R, 0.08571] [enormously, R, 0.08547]

unnecessarily = [unduly, R, 0.22729] [excessively, R, 0.18824]
        [needlessly, R, 0.15204] [extremely, R, 0.14505] [overly, R, 0.14416] [terribly,
        R, 0.13658] [unreasonably, R, 0.13590] [artificially, R, 0.13145]

magnetically = [animatedly, R, 0.09046] [cognitively, R, 0.09002]
        [observationally, R, 0.08789] [inductively, R, 0.08255] [aloof, R, 0.07768]
        [phonemically, R, 0.07564] [irresistibly, R, 0.07373]
        [preferentially, R, 0.07195]

forever = [anyway, R, 0.17271] [again, R, 0.16593] [sure, R, 0.15661]
        [ago, R, 0.15433] [straight, R, 0.15055] [afterwards, R, 0.14857]
        [somewhere, R, 0.14835] [long, R, 0.14649]

sharply = [dramatically, R, 0.21017] [angrily, R, 0.20766]
        [impatiently, R, 0.20760] [harshly, R, 0.19966] [steeply, R, 0.18348] [abruptly,
        R, 0.17989] [steadily, R, 0.16924] [gently, R, 0.16814]

secretly = [privately, R, 0.16576] [afterwards, R, 0.15842]
        [reportedly, R, 0.15368] [doubtless, R, 0.15100] [silently, R, 0.15053]
        [personally, R, 0.14605] [elsewhere, R, 0.14418] [overseas, R, 0.13923]

directly = [indirectly, R, 0.31902] [immediately, R, 0.22114]
        [readily, R, 0.21753] [furthermore, R, 0.21076] [abroad, R, 0.20951] [separately,
        R, 0.20611] [specifically, R, 0.20371] [actually, R, 0.20258]

so = [very, R, 0.48265] [as, R, 0.48239] [quite, R, 0.45538] [too, R, 0.42528]
        [rather, R, 0.40742] [about, R, 0.30683] [more, R, 0.20961] [up, R, 0.13111]

hard = [furiously, R, 0.24553] [again, R, 0.20712] [violently, R, 0.20528]
        [vigorously, R, 0.19598] [forward, R, 0.19058] [backwards, R, 0.18986] [angrily,
        R, 0.18154] [fiercely, R, 0.17672]

dully = [sombrely, R, 0.24398] [despairingly, R, 0.23390]
        [despondently, R, 0.22307] [dumbly, R, 0.21246] [worriedly, R, 0.20896] [bleakly,
        R, 0.20171] [mournfully, R, 0.19973] [dazedly, R, 0.18539]

uncertainly = [expectantly, R, 0.29343] [warily, R, 0.28501]
        [thoughtfully, R, 0.28486] [hesitantly, R, 0.24722] [uneasily, R, 0.24021]
        [disdainfully, R, 0.23756] [admiringly, R, 0.23276]
        [disparagingly, R, 0.21191]

regularly = [frequently, R, 0.22510] [afterwards, R, 0.20536]
        [annually, R, 0.20133] [rarely, R, 0.19901] [occasionally, R, 0.19760]
        [constantly, R, 0.19293] [recently, R, 0.19121] [once, R, 0.18729]
```

Appendix V - Project Logs

Date: 16/10/2006
Details: Defined project focus and span.

Date: 24/10/2006
Details: Discussed system structure,
Similarity measures:
- tree (cyclic, one-way),
- ordered lists of near synonyms,
- comparing sets of words, how many appear in both
- match query words with wordnet categories

Set task: find unix machine in uni on which I can run RASP, consider what information to disregard from RASP output

Date: 31/10/2006
Details: Introduction to RASP and the structure of its output.
Went over some of the RASP output definitions.
Set Task: first version of RASP output filter

Date: 10/11/2006
Details: Session with student support to go over / clarify some English grammar issues

Date: 16/11/2006
Details: Went over filter output, conclusions:
- Canonical forms
- notice passive form & other forms RASP flags.

Discussed interim report content
Set Task: Look at existing solutions for determining word similarities and for question answering systems. Write first draft for interim report.

Date: 29/11/2006
Details: Discussed the methods for evaluating word similarities.
Went over first draft of interim report.

Date: 12/12/2006
Details: Went over ways of matching grammatical relations.
Discussed how to judge the results – precision vs. recall.
Looked at the algorithm and output of the filter.

Date: 22/1/2007
Details: Analysed system output.
Decided on using Diana McCarthy's neighbour data.

Date: 4/2/2007
Details: Considered the various Rasp output formats and clarified miss-understandings about it.
Discussed what to do about words with no neighbour data.
Set task: start collecting texts, considered product reviews.

Date: 15/2/2007
Details: Considered normalizing the score in terms of the query.
Decided to evaluate in comparison between this method and a Google-like search method.
Set task: read through texts and think of questions.

Date: 20/2/2007
Details: Decided to add the original word to word set.
Clarified some terms.
Went over system output noticing problems and possible improvements.

Date: 27/2/2007
Details: Considered structure and content for the report.
Looked at system outputs and analysed its behavior.

Date: 6/3/2007
Details: Tried to solve some scripting problems.
Discussed the various evaluation methods to use on the system output.
Considered minor changes to the system to assist evaluation.
Set task: put together questions which would have answers in particular texts and demonstrated good and bad aspects of eth system.

Date: 13/3/2007
Details: Recognized limitations of the system as a result of initial evaluation.
Discussed how to evaluate the system with respect to precision and recall.
Set task: look at British National Corpus to see what kind of data it holds.

Date: 19/4/2007
Details: Went over the evaluation results to discuss what they say about the proposed method. Discussed some last minute issues regarding the report content.

Appendix VI - The Code

run.sh

Used to compare a user query with the database texts using both comparison methods.

```
echo running query through rasp parser...
echo $*
echo $* | /home/teach/NaturalLanguageProcessing/rasp/rasp3-
            beta_binary/scripts/rasp.sh > parsedTexts/query.txt EOF
java comparator.ComparisonController
```

prepareTexts.sh

The first stage in preparing the database texts to a state in which they can be quickly accessed during the comparison process.

```
for X in *
do
      sh /home/teach/NaturalLanguageProcessing/rasp/rasp3-
            beta_binary/scripts/rasp.sh -m < $X >
done
java comparator.PrepareDatabase
```

runQuestions.sh

Used in order to accelerate the testing by running the test questions one after the other rather than one by one manually.

```
#!/bin/bash
while read line
do
sh ./run.sh $line
done<questions.txt
```