A Statistical Measure of Text Similarity

Peter Russel Dreisiger

This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
Department of Computer Science
The University of Western Australia
1997

Abstract

Identifying semantic similarities using probabilistic language models has received very little attention in the literature to date. While some articles have investigated the ability of statistical language models to generate semantic clusters, these models are traditionally restricted to word prediction and correction applications. This thesis investigates the suitability of using a probabilistic language model to provide an indication of conceptual similarity. The system uses mutual information-based spreading activation to provide a quantifiable measure of the similarity between two passages of text. Although not an absolute measure, it may be used to identify and rank semantically related passages.

Results show that the system is able to correctly group a set of parallel passages found in the Bible, as well as identify paraphrased and simple metaphoric passages. Experiments have also shown that the similarity rankings generated by the program reflect the human perception of similarity.

Keywords: Natural language processing, statistical text analysis, spreading activation, mutual information, meaning extraction, text similarities
CR Categories: I.2.7 [Artificial Intelligence]: Language Models, Language Parsing and Understanding, G.3[Probability and Statistics]: Statistical Computing

Acknowledgements

I would like to thank my Honours supervisors Dr. Mike Kalish and Dr. Paul Hadingham for their helpful comments and for guiding me throughout the year. I am particularly indebted to Mike Kalish for the linguistic and statistical advice he has given me. Special thanks must also go to my parents, who have always been there to support me and provide encouragement.

I would also like to thank Bill, Doyla and Karen Bramblet, and Rod Griffin for putting up with my incessant ramblings, and for participating in one of my experiments. I also appreciate the programming and Linux advice Dr. Chris McDonald has provided me with throughout the year. I wish to thank my fellow Honours students for a great year and for all those amusing incidents. Thanks must also go to James Roelofs for his continual interest in my project, and for his suggestions.

Finally, and most importantly, I wish to give praise to my Creator and Redeemer, the Lord Jesus Christ, whose grace has allowed me to complete this paper, and who deserves the credit for the work described herein.

"The fear of the Lord is the beginning of knowledge..."
Proverbs 1:7

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

The vast amount of online information that has been accumulating over the past decades has prompted the need for new methods of searching and automatically categorising natural language documents. To this end, there have been numerous attempts to develop a system which can identify the underlying semantics contained within a text.

There are many potential areas where natural language research can be applied. These include the following:

- The intelligent retrieval of natural language documents. This would allow a search by meaning rather than by keyword;
- The ability to automatically classify large text databases by topic. If complete document searches are not feasible, the automatic identification of keywords would simplify further retrievals; and
- The application to human-computer interfaces. Despite interfaces designed to allow more user-friendly interaction, current systems are still unable to deal with ambiguous commands.

Representing the meaning of natural language documents has long been the primary focus of natural language research. Although there have been many significant developments in this field, the ability to accurately identify a text's meaning continues to elude researchers [1, 17]. Within computational linguistics, rule-based systems have dominated the literature to date, although recently there has been a growing interest in the use of probabilistic language models [9].

The principal difference between these two approaches lies in how they model the structure of text. Rule-based systems use a phrase's syntactic features to generate a propositional representation which identifies the objects and actions described in the text. In contrast, statistical models analyse the arrangement of words, or surface-forms, within a corpus.

Although representing a document's propositional arguments reflects how human reasoning is believed to work, there are some major problems inherent to rule-based systems [17]. The first is that these systems require vast amounts of hand-crafted knowledge before the system can actually process a passage of text. Even for documents with a limited range of topics, the amount of information required becomes prohibitively large. For this reason, rule-based systems are only effective when dealing with documents within a restricted domain.

Secondly, the use of grammatical rules during the processing stage means that any deviation from the defined language model can cause the program to misinterpret a passage's intended meaning. Depending upon a system's implementation, such variations may even cause the passage to be ignored altogether. Thus, rule-based systems are also restricted in the style and structure of documents that can be processed. Another disadvantage of needing hand-crafted knowledge is that before a rule-based system can be applied to a new language, the syntactic and semantic rules have to be redefined.

For these reasons, the past several years have seen a departure from rule-based systems to hybrid and probabilistic models. Traditionally, statistical language models have been restricted to the correction and part-of-speech tagging of documents [8, 7]. While these models do not consider the propositional structures, or deep-forms, within a document, research has shown that statistical analysis can be used to identify semantically related words [8]. There has, however, been very little research which has looked at using statistical models to identify similarities in meaning.

It is the investigation of statistically-based similarity measures that is the primary focus of my research. By developing a text analysis system based entirely upon word co-occurrences, I will examine the extent to which a text's surface-form can be used to evaluate semantic similarities. The main reasons for choosing to use a statistical language model are:

1. The limitations in scope and grammatical structure inherent to rule-based systems are significantly reduced;

2. The system is self-learning. All knowledge of semantic relationships are acquired through the analysis of large corpora;

3. It is possible to identify semantic clusters without the need to derive propositional representations of the text [8, 11]; and

4. The ability to learn the semantic structure automatically means that statistical models are language-independent.

As no existing systems could be found on which to base this research, the program, and many techniques used to evaluate semantic similarities had to be developed during this project. The approach finally chosen uses mutual information-based spreading activation to perform semantic comparisons, and is implemented in several stages. The first involves compiling word co-occurrences into a statistical knowledge base. For this "language model" to accurately identify semantic relationships, the corpus used to train the system has to represent a significant subset of the English language. The corpus chosen for this project was the King James version of the Bible, as its large scope and vocabulary satisfies this criterion.

After the word co-occurrences were acquired, mutual information was used to calculate the interdependence of each word-pair. The statistical significance of the relationships identified here were then used to compare surface-forms within the training corpus. To provide a measure of the similarity in meaning, a mutual information-based weight was used to represent the strength of semantic associations between words. Spreading activation was then used to identify context-sensitive relationships, and the vocabularies of two phrases were compared to provide a measure of semantic similarity. This technique was then used to allow conceptual, rather than keyword matches to be identified in a search corpus. It is the derivation and performance of this approach that is described in this dissertation.

Chapter 2 reviews the necessary statistical and information theories. A brief overview of spreading activation, and its application to concept exploration is also provided here. This is followed by a review of natural language research in Chapter 3. The use of mutual information in statistical text analysis, and how it can be applied to evaluating semantic similarities is also discussed. Implementation details and considerations are discussed in Chapter 4, along with system performance and requirements. The acquisition of the language model is also described here.

The remainder of the dissertation discusses experimental results aimed at comparing different levels of statistical language modelling. Initially, only mutual information is used to identify similarities. Concept exploration using spreading activation is then investigated. Chapter 6 examines how successfully word energies can be applied to evaluating phrase similarities, and includes an analysis of the self-verification tests. Results of the computer-human comparisons are analysed to determine how closely the computer-generated rankings reflected those generated by human experts are also discussed here. The sensitivity of the model to parameter variations are also investigated.

CHAPTER 2

Probability Theory and Spreading Activation

The statistical analysis of natural language documents can be used to identify structural and semantic patterns within a language. This project compiles the patterns of word co-occurrences (collocations) into a knowledge base. Given the wide range of notation and terminology used in natural language research, the key concepts and definitions are introduced here.

2.1 A Review of Discrete Probability

Probabilistic language models are discrete probability distributions that describe the likelihood of a word sequence occurring in a body of text. These distributions are discrete in nature as every word encountered must correspond to exactly one term in the vocabulary.

Throughout this discussion, it is assumed that the probability of a word-pair occurring can be estimated from its *prior* number of occurrences. Suitable counts are obtained by analysing a sufficiently large corpus. What follows is a brief review of the probability theory used in this thesis.

Given a vocabulary v, the probability of the word w_i occurring is defined as

$$Pr_v(w_i) = \frac{\text{number of occurrences of } w_i}{\text{total occurrences of all words in } v},$$

and the likelihood of it occurring in a sequence of k words is given by

$$P_v(w_i) = k \cdot Pr_v(w_i). \tag{2.1}$$

As this research investigates the statistical relationships *between* words, the language model must also take word co-occurrences into account. This is achieved through the use of joint probabilities. Given the occurrence of w_j in a set of k words, the probability of the sequence (w_i, w_j) occurring is then defined as

$$P(w_i, w_j) = P(w_i|w_j) \times P(w_j), \tag{2.2}$$

where $P(w_i|w_j)$ is called the *conditional probability* and denotes the likelihood of w_i occurring given the occurrence of word w_j.

The last definition used in this project is that of statistical independence. Occurrences of the word pair (w_i, w_j) are said to be *independent* if

$$P(w_i, w_j) = P(w_i) \times P(w_j),$$

or, in terms of conditional probability,

$$P(w_i|w_j) = P(w_i). \tag{2.3}$$

Due to the way in which the language model is constructed, all following equations are expressed in terms of conditional probability.

2.2 Information Theory and Mutual Information

When analysing discrete systems, a measure of statistical interdependence is often used to describe the relationship between two events. In natural language processing, this corresponds to quantifying the statistical interdependence of the words w_i and w_j.

One measure which reflects the entropy of a system is called *mutual information.* It can be thought of as quantifying the *information provided about the event X given the occurrence of the event Y*. For a thorough discussion of mutual information, the reader should consult Gallager [15].

In terms of word-pairs, mutual information is defined as

$$\mathcal{MI}(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)},$$

where the base of the logarithm is arbitrary. Expressed in terms of conditional probability, this becomes

$$\mathcal{MI}(w_i, w_j) = \log \frac{P(w_i|w_j)}{P(w_i)}. \tag{2.4}$$

Using the definition of independence given in Equation 2.3, we can see that a word-pair's mutual information will tend from negative infinity to positive infinity as the pair goes from being mutually exclusive to equivalent. The mutual information will equal zero when the pair is statistically independent.

2.3 Associative Memory and Spreading Activation

Psychologists have long recognised that aspects of human memory are associative, rather than hierarchical in nature. One commonly used model of how such conceptual associations are made is called *spreading activation* [2].

Mathematically, spreading activation involves the propagation of some form of 'energy' across an associative network, whether it is semantic or numeric in nature. When analysing natural language documents, this energy can be thought of as a measure of semantic similarity. In this case, after the activity has been propagated, nodes with high energy levels are likely to have similar meanings to each other [12].

If we treat each concept as a unique node within the network, there are two ways in which it may become active [3]. The first is through external stimulation– some event or occurrence causes a direct activation of the node. In a text processing system, this corresponds to a search word being activated by the user. The second way in which a node may become active is through the spreading of activation. In psychological models of memory, the activity of any given node decreases over time. This is partly due to the nature of short-term memory, but it is also a result of the node's activity being spread to its neighbours through weighted links. In a computerised model, this is simulated by an active node passing part of its activation to its immediate neighbours.

An important distinction between the psychological model of spreading activation and its computerised counterpart is that the latter usually occurs in discrete time. Also, if we reset the activity of all nodes after each search, the need for energy decay is removed, implying that the total energy of the system is conserved. These assumptions serve to simplify our computerised model as follows. Defining $E_n(i)$ to be the energy of node i at iteration n, and $W_{i,j}$ to be the weight of the link joining nodes i to j, the spreading of activation is described by Equation 2.5.

$$E_{n+1}(i) = \rho E_n(i) + (1-\rho)\sum_{j\neq i} E_n(j) \times W_{i,j}. \tag{2.5}$$

The first term accounts for the transfer of energy from a node to its neighbours. Here, ρ is called the *inertia* of the network, and defines how much of a node's total energy will be retained between successive iterations. Thus, over time, the proportion of a node's initial energy undergoes exponential decay. The second term represents the energy received by node i from its neighbours. The weight function used in this project is described in Chapter 4.

To best illustrate the algorithm, let us consider the simplified network shown in Figure 2.1. In this example, the activity of each node is proportional to its shading. The variation in activation levels is a result of the energy being propagated through links of differing weights.

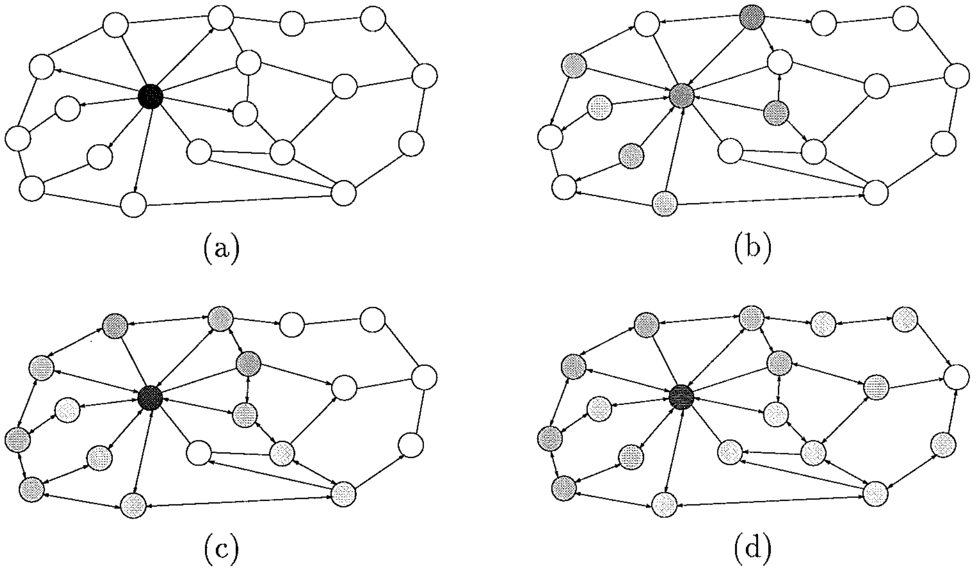


Figure 2.1: Spreading activation through a network over two iterations; (a) initial state of network, (b) energy distribution after first iteration, (c) energy distribution at final state, and (d) active nodes after thresholding.

The network shown in Figure 2.1(a) only has one active node. This corresponds to when we are trying to find associations for a single search word. During the first iteration, this node's energy is propagated according to Equation 2.5. After the first iteration, six new nodes have been activated to varying degrees. The directed arrows in Figure 2.1(b) shows which nodes will become active in the subsequent iteration. Figures 2.1(c) and (d) show the state of the network after two and three iterations respectively. Note that most nodes are now activate. Additionally, the initially active node has regained some of its propagated energy.

The process of identifying associated terms through the spread of activation is often referred to as *concept exploration* [12]. While this process can still be performed given only one activated node, the normal procedure is to apply spreading activation to a set of active nodes. When applied to text analysis, each node corresponds to a word or concept. By spreading the activation across the network, energy is accumulated by semantically related words. Thus, this process can be used to identify context-specific relationships.

CHAPTER 3

Developments in Natural Language Processing

Much of the natural language research completed to date can be classified as either rule or statistic-based. Although rule-based systems are better suited to representing semantic propositions, their dependence upon large amounts of hand-crafted knowledge, and their susceptibility to ambiguous interpretations has prompted an increased interest in hybrid and statistical language models. In this chapter, I review the concepts behind these models, the advantages they offer, and their limitations. Two concept exploration systems based upon spreading activation are also discussed.

3.1 Propositional Representations of Text

The most common method of representing knowledge in computers involves the use of *propositional representations* and *frames.* When used together, they provide the framework needed to combine facts from a knowledge base with the assertions contained in the document being analysed. Propositional representations use logical expressions to denote relationships between the objects and agents described in a passage of text. Frames are then used to combine these relationships into a generalised situation.

In his review of knowledge representation schema, Minsky defined frames as a "collection of questions to be asked about a hypothetical situation. [They] specify issues to be raised and methods to be used in dealing with them" [19]. By providing generalised responses to these questions, frames offer a stereotypical way of defining the world. These stereotypes then allow the computer to make default assumptions about a situation, and to some extent, they even provide the background knowledge required for common-sense reasoning [19].

One restriction faced by propositional systems is their need for this information to be hand-coded into a knowledge base before any text analysis can be performed. Indeed, Minsky acknowledges that even a "minimal common-sense system must know something about cause-and-effect, time, purpose, locality, process, and types of knowledge". Thus, coding the amount of knowledge required by a representational system can become a serious problem; for even a limited domain, the time taken to encode this information makes knowledge acquisition a significant part of a system's development phase [17].

Even if a suitable representation and inference scheme could be found, the current dependence upon humans during the knowledge acquisition stage will continue to be a serious limitation. For this reason, propositional systems are generally restricted in their scope. However, once a set of rules and hand-crafted knowledge has been created, these propositional, or rule-based, systems have been shown to perform remarkably well within their limited domains [17].

One such system was developed by Brachman, Fikes and Levesque [5]. Their KRYPTON knowledge representation program consists of two components; a terminological representation, which is used to define relationships between terms, and an assertational representation which allows assumptions and theories relating the terms to be constructed. In addition, the system also uses manually-defined semantic distances to describe the strength of relationships within the system. These distances are then used to evaluate a total score for each sentence represented, and where there are more than one possible representations, the scores are used to chose the least ambiguous interpretation.

To evaluate the performance of their system, Brachman, Fikes and Levesque investigated its ability to correctly interpret passages of text relating to rock structures. Although their results showed that a rule-based systems can be used to accurately identify a passage's meaning, the limitations of propositional systems were also highlighted— while their system was capable of representing arbitrary concepts, the cost of hand-crafting an entire knowledge base meant that their analysis was restricted to a limited domain.

An example of a system designed to operate over a large knowledge domain is currently being developed by Cycorp [13]. Called Cyc, this system aims to remove domain restrictions by hand-coding vast amounts of knowledge about "everyday life". This ranges from facts and rules of thumb, to knowledge about reasoning itself. While this ambitious project has produced some encouraging results, it is important to note that it has been in development since 1984, and even today the knowledge base is still incomplete. Also, the size of the knowledge base, and the processing requirements of the inference engine mean that the system is restricted to running on powerful computers with large amounts of memory [13].

To circumvent some of these limitations, proponents of rule-based systems have suggested the use of automated knowledge acquisition. Unfortunately, this introduces the need for rules about knowledge, or meta-knowledge [14], which is by no means a trivial requirement. Thus, for systems with more than a limited domain, the work involved in defining semantic rules often becomes a serious issue.

3.2 Hybrid Systems

Recently, some researchers have used statistical language models to augment rule-based systems in an attempt to remove some domain restrictions. These hybrid systems use statistical models to aid in the acquisition and application of rules. Two such systems are reviewed here.

The first was developed by Velardi, Pazienza and Fasolo [20]. Although their system uses propositional representations to describe the semantic structure of a document, the linguistic rules it uses are acquired through the statistical analysis of a large corpus. This is achieved by compiling the probabilities of all word co-occurrences found in the training corpus. For each word-pair, a weight is assigned based upon the strength of its statistical interdependence, and rules are then associated with significant co-occurrences. Finally, these are generalised by manually grouping equivalent computer-generated rules together.

While their system reduced the need for manually generated knowledge, their system still suffered from a need for human interaction. Not only were the semantic weights manually adjusted during the training phase, but final approval of all rules by a linguist was also required.

An interesting system which uses word co-occurrence patterns to simplify the construction of propositional rules was developed by Kavanagh [16]. Rather than constructing a language model, it searches through a corpus and identifies terms which consistently occur near each other. These associations are then presented to a linguist who decides which relationships should be recorded in the knowledge base. The ability of this system to speed up the knowledge acquisition phase also shows that semantic associations are often reflected in word co-occurrences.

While her "Text Analyzer" only used word-pair counts to identify possible associations, part-of-speech tags were also used to improve its effectiveness. Kavanagh found that part-of-speech tags can be used to identify noun-phrases within a document. As these collocations generally correspond to names, this allowed the system to treat them as atomic objects rather than individual words. By identifying co-occurrence patterns at the *conceptual*, rather than word level, Kavanagh found that the system was then able to suggest more relevant associations.

3.3 Statistical Language Representations

Despite decades of research, both rule-based and hybrid systems still suffer from domain restrictions. For this reason, there has been a growing interest in the use of purely statistical language models over the past decade. An important difference between rule-based systems and their statistical counterparts lies in how the text is analysed. Unlike rule-based systems which attempt to construct a representational model of meaning, statistical systems restrict their attention to differences in word structures. While this means that they are unable to compare the underlying semantics of two passages, they can still be used to identify word-groupings, or surface forms. As related texts generally have a similar vocabulary, it may then possible to estimate similarities in meaning based upon these surface forms.

For a statistical representation to correctly identify semantic relationships, a method of analysing the training corpus is required. One technique that has dominated statistical linguistics over the years is the *n-gram language model* [9]. This model makes the assumption that the choice of any given word is only determined by its $n-1$ preceding words, and thus the probabilities of word sequences can be derived by studying a large enough corpus. When $n=2$, these word sequences are called *bi-grams.*

Traditionally, n-gram models have been used in correcting the noisy output-texts that are typically generated by speech and optical character recognition systems. While such systems do not attempt to analyse the underlying semantics of a document, there are some techniques which can be used to identify semantic clusters.

In an attempt to reduce the size of a bi-gram representation of the English language, Brown et al. used a class-based model of natural language [8]. By identifying clusters of statistically dependent terms, they were able to reduce the language model from a word-based bi-gram model to a class-based one. An important discovery made by Brown et al. was that this reduction can be performed with a minimal loss of information. They found that when working with word-pair co-occurrences, this loss of information was minimised when new words were merged into the class whose average mutual information would be maximised. These findings show that statistical text analysis is able to identify semantic similarities within a language.

While their primary motivation was to reduce the size of the language model, the techniques used to generate these word-classes can also be used to identify terms with similar meanings. To construct their semantically-related classes, Brown et al. used a modified bi-gram model which recorded co-occurrences in the vicinity of a word, rather than between adjacent word-pairs. They found that while traditional bi-grams can be used to identify noun-phrases, as the neighbourhood being analysed was expanded, the type of relationships identified changed from syntactic to semantic in nature. As this feature forms an important part of my project, the specific implementation details are deferred until Chapter 4.

Although higher order n-gram models provide a more accurate representation of a language, the space required to store these models increases exponentially with n. In addition, a significantly larger training corpus is required to provide sufficient occurrences of all common word-sequences. While Brown et al. only needed to store the statistics for 1000 word-classes instead of the 260,000 terms in their vocabulary, generalising to a higher order n-gram model means we can no longer use mutual information alone to minimise a word-class' loss of information [8].

Also, the number of classes would have to be increased considerably, as shown by an investigation of a tri-gram based system. During its training phase, the Tangora speech recognition system developed by IBM encountered over 90 million different tri-grams [4]. Out of these, it was found that only 12 million ever occurred more than three times in a 365 million word training-corpus. Thus, it is apparent that anything larger than a bi-gram language model can also become extremely inefficient.

Traditionally, statistical language models have been used in applications where the wording, rather than the meaning of a phrase is being analysed. Even though they are unable to represent propositional structures, statistical analysis has been used to identify semantic relationships, however no literature describing a statistic-based similarity measure could be found.

3.4 Concept Exploration using Spreading Activation

Thus far, only methods of constructing language models have been discussed. In this section, the use of spreading activation in concept exploration systems is examined. Traditionally, activation is spread over a semantic network, where link-types are used to determine the flow of activity. However, one system reviewed here uses a hand-weighted network to bias the associations defined by the semantic network [10].

The GRANT expert system developed by Cohen and Kjeldsen [12] is an example of a system which uses spreading activation over a semantic network. It was designed to find suitable matches between proposed research projects and funding organisations. To this end, information about each available grant, along with semantic relationships between different research fields was coded into the network. After receiving a search request from the user, the system attempts to find a direct match between the project and an available grant. If no match can be found, then related subjects are identified using spreading activation. This system functioned well within its domain, but was limited by the need for semantic relationships to be hand-coded.

As their system was based on a semantic network, Cohen and Kjeldsen were able to use the link-types to restrict the spread of activation to instantiations rather than generalisations. One example they used to justify this constraint was that while two research topics may both be related to the concept *science*, this does not guarantee the both will receive funding from the same organisation. Based upon this, and the fact that general terms often have a large number of connections, two other restrictions were applied:

- A *distance* constraint so that the activation-passing will cease after a finite number of links (Cohen and Kjeldsen used a limit of four); and

- A *connectivity* constraint so that the activation should cease at nodes with a high degree of connectivity.

Intuitively, related terms would generally share a common neighbourhood with each other. It was found that limiting the number of traversals reduced the computation time without adversely affecting the system's performance.

After spreading the activation over a fixed number of links, the newly activated words were then used to expand the database search. Although the final system did not take into account the search phrase's syntactic structure, they found that a partial, word-only match was able to identify most of the associations considered relevant.

In contrast, the concept exploration system developed by Chen, Basu, and Ng [10] spread the activation over a heterogeneous group of networks. While traditional semantic networks were used to describe the relationships between concepts, a numeric network was also used to describe the strength of these associations. When searching through document abstracts, they found that the inclusion of semantic weights resulted in the system identifying an increased number of relevant associations. Although their system was based upon hand-crafted knowledge, the inclusion of weighted links can be generalised to implementations of spreading activation over purely numeric networks.

Their research also investigated the differences between a serialised branch-and-bound implementation of spreading activation, and a Hopfield network-based parallel relaxation implementation. The parallel relaxation approach found words with global energy maximas, whereas the branch-and-bound technique iteratively propagated energy between nodes. Interestingly, they found that both methods produced similar results, although the parallel approach was far more computationally intensive.

One additional requirement the serial method had over parallel relaxation was the need for a stopping criteria; while the Hopfield network converged to the optimal solution, the serial approach required a user-specified stopping condition to terminate the spread of activation. As the branch-and-bound method used an iterative approach, after a certain number of iterations, the entropy of the system would become so large that all nodes within the network would have a comparable activation. In their system, Chen, Basu, and Ng used a user-specified constant as the stopping criteria.

Although the spread of activation has been successfully implemented on semantic networks, its application to probabilistic language models has gone largely uninvestigated. Using the mutual information generated by the modified n-gram model, this research investigates the suitability of spreading activation over a statistical knowledge base. The ability to identify similarities between phrases is also investigated.

CHAPTER 4

The Statistical Text Analyser

The statistical text analyser written for this project uses word co-occurrences to detect semantic associations. These relationships are identified by analysing the mutual information of word-pairs drawn from a training corpus. As a statistical knowledge base acquires all its information through training, a sufficiently large corpus is required to provide an accurate representation of the English language. The King James version of the Bible was chosen for this project as it satisfies both of these requirements; firstly it is over 750,000 words long, and secondly it has a vocabulary of more than 12,000 words.

This program implements the modified n-gram language model used by Brown et al, [8] in which a sliding word-window is used to define the concept of a word neighbourhood. Terms within this window are said to be *near* each other, and co-occurrences between these words and the central term are recorded. The resulting "language model" is then used to calculate the interdependence of word patterns, and through the spreading of activation, context-sensitive associations can be identified.

Due to the lack of literature investigating the use of a statistical similarity measure, all the code described in this chapter had to be written from scratch. This chapter details the specific theory involved in the design of the system, and discusses the techniques used to construct the knowledge base from the training corpus. System performance and scalability are also examined.

4.1 Knowledge Acquisition

The first stage of the text analysis process involves identifying the semantic patterns present in the training corpus. Prior to constructing the statistical knowledge base, some pre-processing is performed to reduce the size of the language model, and improve its accuracy.

Once this preliminary processing has been completed, the program then constructs the actual knowledge base in the form of an interconnected network.

4.1.1 Preprocessing

Before a word is added to the knowledge base, up to four separate operations may be performed on it. The first two stages are always carried out, and involve the removal of punctuation marks and stop-words. The second two are optional, and involve mapping all words to their lowercase version and recording their part-of-speech tags.

Removal of punctuation

As punctuation marks are only used to identify grammatical structures, not only does their inclusion in statistical text analysis provide little information, but they increase cause the size of the knowledge base to increase significantly. Since each word may be punctuated in several different ways, their inclusion means that each variation would be represented by a unique node. As each variant would also have a lower frequency count, this would result in decrease in the language model's accuracy. Thus, before any further processing is performed the `!,.:;?()\'''` and `'` characters are removed from the input word.

Stop-word removal

As define by Kavanagh [16], stop-words are the conjunctions used to maintain a passage's grammatical structure. Like punctuation marks, they provide little semantic information at the word-level. During this part of the pre-processing stage, the words shown in Appendix A are replaced by a null-word. As this is only done to maintain the consistency of word co-occurrences, the null-words are ignored during subsequent stages of the analysis.

There are two reasons for removing stop-words. Firstly these words are so common that nearly every other word would be linked to a large set of stop-words. While only increasing the size of the vocabulary of words by approximately 100 words, the inclusion of all their links was found to increase the size of the knowledge base by over 40 percent. Secondly, in addition to providing little semantic value, their high occurrences cause these stop-words to have high values of mutual information. During the spreading activation stage, these words then accumulate most of the network's energy while providing no semantic gain.

Removal of Capitalisation

This is the first of the optional stages. By default all letters are converted to lower case before the processing stage. This allows the program to identify any capitalisation of a word with one node in the network. When the Bible was processed, this yielded a saving in memory of over 40 percent.

Although the capitalisation of words occurring at the start of a sentence provide little semantic information, the occurrence of capitalisations within a sentence may indicate the presence of a specific name or object. In this case, the removal of capitalisation can cause these distinctions to be lost. However, capitalisations alone are not sufficient to differentiate between all such instances, and to compensate for this loss of information, one final stage was added to the program.

Part-of-speech Tagging

In her text analyser, Kavanagh [16] found that a word's part-of-speech tag can be used to identify different instances of the same word. Differences in a word's meaning are usually marked by a differentiation in its part-of-speech tag. For example, the noun *Bob* refers to an person, while the verb *bob* describes an action.

The option of tag analysis is provided by the program through the use of publicly available software. Brill's Part-of-speech tagger for Linux [6] was used to annotate each word in the corpus. While this program can differentiate between over 40 tags, the pre-processing stage maps them down to one of four types being *noun*, *verb*, *adjective*, and *miscellaneous*. Although the program then treats each tag as a separate word, effectively increasing the vocabulary, the program is structured so that common features of each node are shared between tags, thus reducing the amount of space required. Analysis of the King James Bible showed that each word had an average of 1.6 tags, and their inclusion caused a corresponding increase in the size of the knowledge base.

4.1.2 Parsing the Corpus

During this stage, the pre-processed text is now used to generate the statistical knowledge base from word co-occurrences. These patterns are recorded through the use of a sliding word-window similar to the one used by Brown et al. [8]. They found that by analysing co-occurrences within a certain distance of each word, semantic relationships could be identified. In this project, it is the word-window that is also used to define the notion of word proximity.

As we are only interested in semantic, rather than syntactic relationships, we ignore the areas in which conjunctions are most likely to occur. Thus, the cut-off region is defined as the region directly adjacent to the central term. Any word within the window, but outside the cut-off region, is said to be *near* the central word. Since syntactic structure is ignored, a symmetric word-window is assumed for simplicity. Given this symmetry, we can describe any word window in terms of its 'radius' (the furtherest word it includes) and the cut-off radius, as shown in Figure 4.1.

Another consequence of this symmetry is that the mutual information also be-

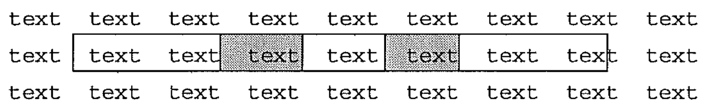

text text text text text text text text text
text text text text text text text text text
text text text text text text text text text

Figure 4.1: Example of a word window with radius 3 and a cutoff radius of 1; a (3,1) word window

comes symmetric. This means that

$$\mathcal{MI}(w_i, w_j) = \mathcal{MI}(w_j, w_i), \tag{4.1}$$

where the mutual information is now defined as

$$\mathcal{MI}(w_i, w_j) = \log \frac{P(w_i \text{ near } w_j)}{P(w_i)}. \tag{4.2}$$

The parsing process itself involves the repetition of two simple steps; first the co-occurrences between the central term and the words *near* it are recorded, then the window is shifted through the corpus one word to the right. This procedure is then repeated until every word co-occurrence in the training corpus has been recorded.

Although it would be quicker to compile these co-occurrences into their mutual informations, this would mean that once the values have been calculated, no further training can be performed. Ideally, the system could continue its training by parsing every new search document or phrase it encounters to improve its language model. To allow the network to be continually updated, only co-occurrence statistics are stored, and the mutual information is calculated from these as needed.

4.1.3 Network Structure

Each time a co-occurrence is recorded, the strength of the link joining the two words is updated. If no such co-occurrence has been processed before, then a new object is added to link the two words together. For the Bible, each word in the vocabulary has an average of 590 such links when processed using a $(10, 1)$ word-window. Since there are over 12,000 words in the vocabulary, this quickly forms a densely inter-connected network.

As each link corresponds to a co-occurrence within the window, we can think of the strength of a semantic relationship as being loosely related to the number of links that have to be traversed to connect two concepts. To provide a better approximation of semantic similarity, spreading activation and mutual information is used to take conceptual neighbourhoods into account.

4.2 Text Analysis

Once the statistical knowledge base has been compiled, the program then uses this information to identify semantic relationships recognised during the training phase. This is accomplished in two stages; the first uses concept exploration to generate a set of related search words, and the second involves searching through a corpus to find the verses which contain the most number of these terms.

4.2.1 Spreading Activation

As described in Chapter 3, spreading activation has been used successfully for concept exploration over semantic networks. The text analyser developed here applies spreading activation to a statistical knowledge base.

Originally, a parallel relaxation implementation of spreading activation was considered, but as Chen et al. [10] found, these results were comparable to those generated by a standard iterative implementations. Given the differences in computational time, the iterative approach was finally adopted.

Some of the constraints used by Cohen and Kjeldsen [12] were also investigated, but of the two described in Section 3.4, it was found that both could be implemented implicitly. The first constraint considered was the *distance*; Cohen and Kjeldsen argued that if two terms are semantically related, they should be connectable by a small number of links. While this is true, limiting the number of iterations over which activation is spread places an equivalent restriction on the system.

The other restriction was the *connectivity* constraint which said that propagation should be terminated at a node with a large number of connections. This was not explicitly coded for two reasons; firstly, with stop-words removed from the corpus, the words with a high degree of connectivity tended to be important conceptual words. Secondly, given the number of outgoing links, the average energy transfered from a general node to its neighbours tended to be negligible.

After each iteration, each node's energy is updated according to Equation 2.5. The choice of the weight function W_{ij} was not immediately obvious since the mutual information used to describe semantic relationships can range from negative to positive infinity, while the link weights must lie between zero and one.

The weight function finally settled upon involves a ratio of mutual information. Each link's weight is determined as a ratio of its mutual information to the total across all links whose mutual information exceeds zero. Algebraically, it is defined as

$$W_{ij} = \frac{\mathcal{MI}(w_i, w_j)}{\sum_k \mathcal{MI}(w_i, w_k)}, \tag{4.3}$$

for all k's resulting in a positive value of $\mathcal{MI}(w_i, w_k)$.

The initial activation of words within the knowledge base is restricted to search words the user has specified. By propagating energy from these search nodes, the text analyser can identify related concepts which can aid in the search process. The program also supports the option of giving a node a *negative* initial energy which will cause that word to have an inhibiting effect on the spread of activation. This is similar to the logical **not** operation supported by most search engines.

It should be noted that while the mutual information between two words is symmetric, this is rarely the case with link weights. As general terms have a large number of links, the proportion of energy they pass to their neighbours is less than the converse. Thus, while a specific term will identify its generalised description, the activation will effectively cease at the general node.

While psychological instances of spreading activation use energy decay to forget rarely used concepts in short-term memory, this temporal decay is not implemented in my program. Instead, after each search, the activity of all nodes within the network is reset, allowing a completely new search to be performed.

4.2.2 Similarities

Once the network has been activated by the search phrase and the associations identified, the similarity between two verses can then be evaluated. Originally, a sliding word-window was used to evaluate similarities, but overlaps between verses adversely effected the system's performance. Kavanagh found that automatically delimiting sentences can prove to be a problem when a text contains abbreviations and quotes [16], so it was decided to make use of the verse delimiting provided with the Bible. Accordingly, the verse was chosen as the fundamental topical unit, and comparisons were then made on the verse level. Since stop-words were ignored during the training phase, they were also ignored here, and so their inclusion in search or matched phrases has no bearing upon their similarity score.

Due to resource constraints, only a queue of the most similar verses is maintained by the program, and as a new verse is found with a higher score higher, it is inserted into the list while the least similar verse is dropped. In all experiments, a list of the top twenty phrases was maintained.

Originally, the total energy of each unique word was used to define the similarity, but this resulted in many unrelated verses being selected. An inspection of search results showed that longer verses, by virtue of their number of unique words, dominated the similarity lists. To correct this, the scores were normalised according to length, and after some experimentation, it was found that the following formula gave a satisfactory measure of similarity:

$$Similarity = \frac{\sum_{w_i \in v} E_i}{\sqrt[3]{\text{verse length}}}. \qquad (4.4)$$

Using this technique it is possible to find the most similar matches in the entire Bible in less than 40 seconds.

4.3 Program Execution

This project was implemented in C++ and has been compiled on a 6x86-200+ PC compatible with 40Mb of physical RAM and 360Mb of swap space. The machine was running Linux 2.0.0 and used the GNU C++ compiler version 2.7.2. Comparisons in program resources are provided in the following section, as is an example of the program's user accessible features

4.3.1 System Performance

After several revisions, the code was optimised to the stage where typical training and retrieval sessions took approximately five minutes. A list of system resources for different system configurations is shown in Table 4.1.

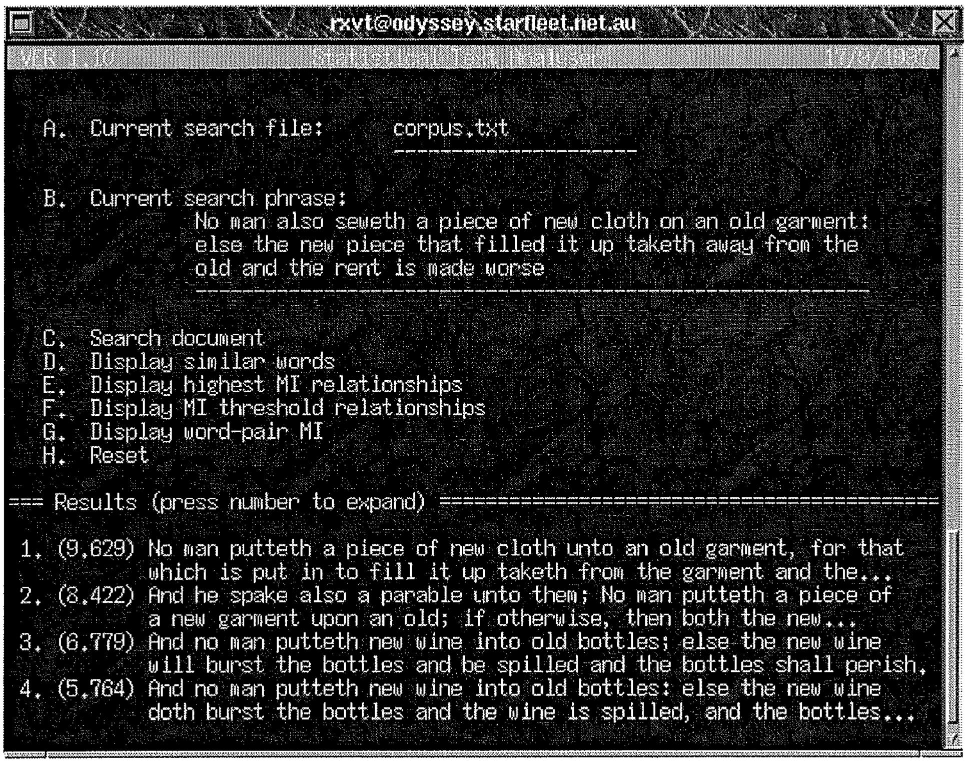
Configuration	Lines of text	Words	Vocabulary	Time	Memory
$(250, 10)$	20000	194718	5497	46:24	26Mb
$(250, 10)$	40000	386348	8792	2:19:28	50Mb†
$(250, 10)$	82871	789639	12605	11:47:17	93Mb†
$(10, 1)$	20000	194718	5497	1:50	4.5Mb
$(10, 1)$	40000	386348	8792	2:50	8.6Mb
$(10, 1)$	82871	789639	12605	5:15	16Mb
$(10, 1)$ + capitals	82871	789639	12605	8:01	23Mb
$(10, 1)$ + pos tags	82871	789639	12605	8:50	26Mb

Table 4.1: Execution results for various system configurations; *pos tags* refers to a test including part-of-speech tags, and *capitals* refers to the inclusion of word capitalisations. †= cases requiring large amounts of virtual memory. (r, x) is the window of radius r and exclusion radius x.

One can see that there is a huge difference in execution time and memory requirement as the system is switched from a large word-window to the final size chosen for analysis of the Bible (as discussed in the next chapter). Additionally, as long as the entire knowledge base can reside in main memory, the performance of the system is relatively linear with respect to corpus size.

4.3.2 Sample Session

Currently the access to word-pair mutual information, mutual information rankings, spreading activation associations and corpus search features of the program are contained within separate executables. However, it is planned to integrate all this functionality into one executable which can provide with the user with all necessary features. A sample screen-shot of this interface is shown in Figure 4.2.



```
VER 1.10            Statistical Text Analyser              17/8/1997

  A.  Current search file:      corpus.txt
                                ------------------------

  B.  Current search phrase:
                      No man also seweth a piece of new cloth on an old garment:
                      else the new piece that filled it up taketh away from the
                      old and the rent is made worse
                     ---------------------------------------------------------------

  C.  Search document
  D.  Display similar words
  E.  Display highest MI relationships
  F.  Display MI threshold relationships
  G.  Display word-pair MI
  H.  Reset

=== Results (press number to expand) ==========================================

 1. (9.629) No man putteth a piece of new cloth unto an old garment, for that
            which is put in to fill it up taketh from the garment and the...
 2. (8.422) And he spake also a parable unto them; No man putteth a piece of
            a new garment upon an old; if otherwise, then both the new...
 3. (6.779) And no man putteth new wine into old bottles; else the new wine
            will burst the bottles and be spilled and the bottles shall perish.
 4. (5.764) And no man putteth new wine into old bottles: else the new wine
            doth burst the bottles and the wine is spilled, and the bottles...
```

Figure 4.2: Sample screenshot

CHAPTER 5

Mutual Information-based Concept Exploration

We have seen that the statistical analysis of word co-occurrences has been used to identify semantically related groups. The experiments described in this chapter investigate the relevance of these associations, and how their quality changes as we go from a single to dual-word mutual information-based search. The results of spreading activation-based concept exploration are also examined.

5.1 Mutual Information between Two Words

This section investigates the word associations generated using mutual information and the modified n-gram language model described by Brown et al [8]. To compile word collocation statistics, they employed a sliding word-window to record co-occurrences during the training stage.

Although the same approach was used here, the most significant difference lies in the size of the word-window; Brown et al. used a (500,10) sliding window, while the one used here was only (10,1) words long. Initially, a (500,10) and (250,10) word-window was used to analyse the Bible, but the associations generated using these large windows were too abstract to identify any meaningful associations. By examining the mutual information-based associations, it was found that a (10,1) word-window provided the best results.

One possible reason for the large difference in window sizes is the use of different training corpora. While Brown et al. used Canadian parliamentary transcripts in which many pages were devoted to a single topic of discussion, the length of topical units in the Bible rarely exceeds several verses.

This is in agreement with Brill et al. [7] who used mutual information to determine part-of-speech tags. Their analysis of sentence-level structures, revealed that a (10,0) word-window provided optimal results.

After generating the co-occurrence statistics using a $(10, 1)$ word-window, tests were performed to see what semantic associations could be identified. An example of some mutual information-based associations are shown in Table 5.1.

Word searched	*Computer identified associations*
water	waters, troughs, rinsed, bathe, swimmest, lappeth, overfloweth, buckets, barrels, drawer
bread	loaf, wafer, omers, morsel, bakers, cake, feedest, famished, mealtime, mouldy
ship	galley, oars, shipmaster, anchors, landed, aboard
fruit	planteth, prune, gardens, flourishing, yield, leaf
apostle	apostles, preacher, teacher, evangelists, ordained
adulteress	adulterer, adultery, married, husband, wife
servant	bondservant, slave, payment, bowing, zereda
harvest	ingathering, reapest, seedtime, cuttest, labourers
flood	carriest, tempest, overflown, euphrates, desolation
destroy	revengers, impoverished, devour, battlements, windy
fire	torch, fuel, flame, flames, scorch, hot
noise	rattling, stilleth, attentively, commotion, rang
joyful	glad, singing, solitary, clap, roar, aloud
crying	scorneth, murderer, stirred, tears, pain, cry
war	rebellest, weapons, warfare, occupiers, hagarites
sacrifice	delightest, sweetsmelling, goats, thanksgiving
lamb	kid, wolf, ram, nourished, blemish, shearer
life	loseth, immortal, mortality, adventured, days
jericho	plains, pisgah, spy

Table 5.1: Computer identified semantic relations. These words were taken from the top fifteen list of mutual information-based associations.

It is apparent that even using simple mutual information rankings, meaningful associations can be identified. For each search word, the text analyser ranked the related words according to their mutual information. The associations shown in Table 5.1 were taken from the top fifteen ranked words.

While most relationships identified using mutual information are indeed valid, it is interesting to note the *type* of associations detected. As a consequence of analysing the mutual information of word co-occurrences, the majority of the terms matched are words that are found in close proximity to the search word. Although some synonymous relationships were identified, most of the matched words are only related conceptually.

An example of these conceptual relationships are the associations derived for the word *water*. While these words are associated with the concept of water, none of them can be used interchangeably. Also, the fact that Jericho was situated in the plains below Mount Pisgah was reflected in these associations.

Mutual information, by itself, can also generate associations that might not seem obvious. These errors may be a consequence of using mutual information over a limited training corpus; when a less common word frequently occurs around another term, it is assigned a high value of mutual information. One way to reduce these types of associations would be to use a larger training corpus. An example of this the association between the name *zereda* and the word *servant.*

From these results, we can see that although mutual information-based associations can identify some relevant relationships, we need some way to encourage the selection of semantically similar words. One way to do this is by taking into account a node's surrounding topology rather than just its immediate neighbourhood. This corresponds to spreading the activation over more than one iteration.

5.2 Spreading Activation from a Single Word

To improve the identification of semantic associations, the analysis of text should take into account each word's context and relationship to other related words. This can be done by expanding the search from a node's immediate neighbours to a localised region of the network. In this section, the differences between mutual information-based associations and those identified through the spread of activation are examined.

Associations for this experiment were obtained by activating a single search word and propagating the energy through the network. Based upon experiment conducted in Chapter 6, an inertia of $\rho = 0.4$ was used, and the activation was spread over four iterations. Results for the same search words used in the previous section are shown in Table 5.2, with new entries shown in bold.

Although some words were identified by both techniques, we can see that spreading the activation over four iterations has increased the number of synonymous relationships identified. Examples include the association of *rivers, fountain, pools* and *wells* with the word *water.* Similar results were obtained for the other words.

While some identified terms could not be used interchangeably, the majority of these associations have a strong conceptual relationship. Examples include the association of *mealtime* with *bread,* and *nets* with *boats.*

Thus, while both mutual information and spreading activation can identify related terms, the associations found by the spread of activation tend to reflect conceptual, more than semantic relationships. A fact not reflected by these tables is that the difference in energy levels between conceptually related terms and the other associations was greater when spreading activation was used to identify relationships, than when mutual information alone was used.

Word searched	*Computer identified associations*
water	waters, **rivers**, **fountain**, **pools**, **wells**, bathe, **wash**, **drink**, **pour**, **baptizing**
bread	**eat**, **unleavened**, cake, wafer, mealtime, morsel, **loaves**, **flesh**, **wine**
ship	**boats**, galley, **sailing**, oars, **nets**, **coasts**, **tempestuous**
fruit	**fig**, **vine**, **leaves**, prune, planteth, **branch**, gardens
apostle	apostles, preacher, teacher, **profession**, evangelists
adulteress	adulterer, adultery, married, **whore**, **law**, wife
servant	bondservant, slave, **fellowservant**, **commandedest**
harvest	**field**, reapest, **wheat**, **sickle**, seedtime, **sheaf**
flood	**noe**, **storm**, **marrying**, **drowned**, **rivers**, **stream**
destroy	**utterly**, revengers, windy, battlements, **dishonour**
fire	**hearth**, **burned**, flames, **smoke**, **fuel**, **wood**, torch
noise	rattling, **whip**, **rushing**, **billows**, **shout**, rang
joyful	**rejoice**, **shout**, **sing**, **joy**, clap, **thanksgiving**
crying	**madmen**, **infant**, **voice**, tears, **loud**, **pursue**
war	**battle**, weapons, **fight**, **swords**, **armed**, **hagarites**
sacrifice	**offering**, **burnt**, delightest, goats, **rams**
lamb	ram, kid, **young**, **offering**, **pigeon**, **slaughter**
life	**death**, **lose**, **soul**, **eternal**, **lives**, days, **good**
jericho	plains, pisgah, spy, **valley**, **jordan**

Table 5.2: Associations generated using Spreading Activation. These words were taken from the top ten list of associations identified using spreading activation. Words not previously identified by mutual information are shown in bold.

5.3 Concept Exploration using Mutual Information

The first two sections of this chapter have looked at semantic associations generated by one search word. As any comparison of underlying meaning requires the language model to identify context-sensitive relationships, it is ability which is investigated here. Although the next chapter examines the evaluation of similarity across entire sentences, this section examines the extent to which context-sensitive associations can be made using only two keywords.

The results presented here were obtained using mutual information only. By combining two words, we are able to examine how the list of related words reflect the different contexts. Table 5.3 shows the selected words along with their resulting matches.

As expected, the lists generated using mutual information are generally a union of the two individual lists generated earlier. This is because each word has an energy proportional to the weight connecting it to the active node. While some associations are relevant, it is apparent that the system has been unable to provide

Words searched	*Associated words*
water, harvest	overfloweth, gleanings, labourers, banks, sickle
water, flood	carriest, driedest, drowned, overflown, noah
destroy, water	revengers, windy, battlements, bondservice, devourer
destroy, fire	fuel, coal, revengers, battlements, fan, devourer
noise, joyful	psalm, cornet, beautify, clap, sing, joy
noise, crying	headstone, madmen, tumult, infant, scorneth, loud
noise, war	rattling, stillest, commotion, billows, roar, tumult
sacrifice, lamb	shearer, goats, kid, ram, offer, burnt
sacrifice, life	delightest, pertain, thanksgiving, spareth

Table 5.3: Context-sensitive associations generated using Mutual Information.

true context sensitivity, but rather has just merged two individual lists according to their absolute mutual information weightings.

5.4 Concept Exploration using Spreading Activation

The results of spreading activation from two key-words are shown in Table 5.4. These associations were generated over four iterations using an inertia of $\rho = 0.4$.

Words searched	*Associated words*
water, harvest	**field**, banks, **reapest**, **summer**, **ripe**, overfloweth, **rain**
water, flood	**fountain**, **rivers**, drowned, overflown, **storm**, noah
destroy, water	**fountain**, **storm**, windy, revengers, **drowned**, devourer
destroy, fire	**devour**, **quenched**, fan, **burned**, **flame**, **consuming**, **fury**
noise, joyful	**shout**, psalm, sing, **rejoice**, thanksgiving, clap
noise, crying	tumult, infant, madmen, **weeping**, **joy**, **hosanna**
noise, war	commotion, **shouted**, rattling, **whip**, **hoofs**, **stamping**
sacrifice, lamb	**offering**, burnt, ram, **goat**, **slaughter**, **incense**
sacrifice, life	**vow**, **wellpleasing**, **oblatation**, thanksgiving, **death**

Table 5.4: Context-sensitive associations generated using Spreading Activation. Newly identified terms are shown in bold.

Unlike the mutual information-based concept exploration, there is more even representation of terms associated with both keywords. This is shown by the searches involving *destroy*. Here the final energy of terms such as *battlements* and *revengers* have been suppressed, even though they have high values of mutual information, allowing other context-specific terms to be identified.

These associations show that the statistical knowledge base's output is context sensitive; even though each test subset had one word in common, the identified associations were significantly different. As with the one-word experiments, spreading activation resulted in the identification of synonyms as well as conceptually related terms.

By comparing Tables 5.1 through to 5.3, we can see that although there are some words in common with the results shown in Table 5.4, some additional terms were identified. Another factor not shown here is that while there are some words common to all four tables, their energy values varied between experiments. As when the activation was spread from a single word, the difference in the energy of related and unrelated terms was higher when spreading activation was used for concept exploration.

5.5 Discussion

We have seen that while mutual information-based rankings can be used to identify conceptual associations, in both the single and dual keyword experiments, the spreading of activation was able to identify a greater number of equivalent words. While the associations generated using mutual information were valid, they only identified words that commonly occur near the search term. As such, mutual information-based concept exploration is generally limited to returning sets of frequent co-occurrences. In contrast, context-sensitive searches using spreading activation identified concepts which occurred around the same type of words as the search term. Thus, these associations provide a better degree of context-sensitivity.

By extending concept exploration to the point where every word in a search phrase gets activated, we are able to generate a list of context-sensitive semantically related words which can then be used to search a large corpus for a match. This is the basic approach used in the next chapter.

CHAPTER 6

Evaluating Text Similarities

The final stage of my research involved the use of concept exploration to evaluate the similarity between two passages. The first two experiments described here involved entering a search verse, performing the spreading activation and then searching through the Bible to find possible matches.

For each verse, the total energy of all distinct terms was evaluated, and then normalised according to verse length. Each test only considered the top twenty matches and their relation to manual cross-references and human similarity rankings.

6.1 Self-verification

This section investigates the system's ability to identify parallel passages within a corpus. As mentioned in the introduction, one of the reasons the Bible was used in this project was because it contained many such parallels. Within the Bible, most major events are described by at least two people, and as these have been manually cross referenced, it is easy to verify which parallels the program has successfully detected.

The experiment itself attempted to match 25 search verses with their parallel passages. The criteria used to assess this ability was the difference between the number of referenced parallels, and the smallest number of verses which contained them all. Occurrences of an unreferenced verse within this containing set are called *intrusions*, and the number of intrusions each verse has is called the *intrusion depth*. An intrusion depth of > 20 means that at least one parallel was not found in the list, and corresponds to a fail.

Results for zero and one iteration correspond to a keyword and mutual information-based search respectively. The intrusions for $\rho = 0.4$ at seven different iterations are shown in Table 6.1.

As can be seen, the performance of the text analyser dropped off markedly for more than four iterations. Even though most parallels have a number of words

Intrusion	*Number of Iterations*						
Depth	0	1	2	3	4	5	6
0	9	10	13	18	22	4	0
1	4	4	7	4	2	3	1
2	2	5	4	2	1	2	0
3	3	2	0	0	0	3	1
4	1	3	1	0	0	2	2
5	3	1	0	1	0	1	2
6	1	0	0	0	0	2	1
7	0	0	0	0	0	1	0
8	0	0	0	0	0	2	1
> 20	2	**0**	**0**	**0**	**0**	5	17

Table 6.1: Intrusion counts per depth for different numbers of iterations.

in common, the keyword search was unable to identify all of them for two verses. While the mutual information-based search was able to identify all parallels, the extra iteration did not yield a significant improvement over the basic keyword search.

Optimal performance was achieved after four iterations, which only produced a total of four intrusions. In this test, all four corresponded to shorter, summarised versions of the search verse. These shorter verses only out-ranked the true parallels because of the correction for verse length. This is a significant improvement to the 42 total intrusions generated by the keyword search, and the 37 intrusions when only mutual information is used.

From Table 6.1, we can see that the performance of the text analyser drops off rapidly for more than four iterations. Unlike the intrusions found after four iterations, very few of the verses identified after five iterations were actually relevant. One reason for this is that after four iterations, the energy has become so widely dispersed through the network that all word activations become comparable. This may be a consequence of the knowledge base's high degree of connectivity; for typical search phrases, all nodes have been activated after four iterations. Up to four iterations, however, there is a significant energy difference between related and unrelated words.

Changes in system performance were also investigated for variations in inertia. Using the same 25 verses, the experiment was repeated for four iterations as the inertia was varied from $\rho = 0.2$ to $\rho = 1.0$. These results are shown in Table 6.2.

As an inertia of $\rho = 1.0$ means that no energy is propagated between iterations, these results are also equivalent to a keyword search and thus are equal to the results for zero iterations described in Table 6.1. The results of this experiment indicate that an inertia between $\rho = 0.4$ and $\rho = 0.6$ provide the optimal performance. As with the previous experiment, the intrusions under optimal parameter

Intrusion	*Inertia* (ρ)								
Depth	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0	4	22	20	20	16	14	11	9
1	0	1	2	2	1	5	6	3	4
2	1	3	1	2	2	3	2	6	2
3	2	3	0	1	2	1	1	2	3
4	1	2	0	0	0	0	2	2	1
5	2	5	0	0	0	0	0	1	3
6	2	0	0	0	0	0	0	0	1
7	0	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0
> 20	16	6	**0**	**0**	**0**	**0**	**0**	**0**	2

Table 6.2: Intrusion counts per depth as a function of Inertia.

values were caused by shorter paraphrasings of the search verses. The matches generated for values of inertia below $\rho = 0.4$ were mainly comprised of unrelated sentences, as under these circumstances the spreading activation energises words according to their statistical co-occurrences while de-emphasising the original search words.

In general, low values of inertia and high iteration counts mean that the majority of energy associated with the search phrases has been dissipated outside a useful neighbourhood.

6.2 Human-computer Comparisons

This experiment investigates how closely the computer reflects human perceptions of semantic similarity. To do this, four volunteers who are well versed in biblical terminology were used to provide a reference against which the statistical model was compared. The results of these comparisons, along with an investigation of the model's sensitivity to parameter variations are discussed in this section.

6.2.1 Performance of System for Optimal Parameter Values

The experiment was performed in three stages; the first involved compiling a list of computer-generated similarities, the second involved obtaining human similarity ratings for the verses in this list, and the third looked at how these two results compared.

The list used in this experiment consisted of 11 reference verses, each of which had 20 computer-associated verses and four irrelevant verses taken from the Bible. For each reference verse, the order of its 24 corresponding verses was jumbled, and the computer generated scores were removed, although the reference verse was clearly marked.

Each of the four subjects were then given an 11 page booklet listing the 264 verses. For each verse in the 11 lists, they were asked to rate how closely it resembled the meaning of its reference verse. Ratings were given on a scale of zero (corresponding to no similarity in meaning) to ten (the two verses had an identical meaning). It was emphasised that similarities in the underlying *meaning*, and not word correspondences, were to be examined. After these ratings were taken, for each subject, the verses in each of the 11 lists were then ranked according to their similarity rating. It was the relationship between these rankings that was then investigated.

For this experiment to provide any meaningful results, there had to be a reasonable agreement between subjects as to the verse similarities. Thus, the first stage of the analysis investigated the inter-subject correlations, which are shown in Figure 6.1. The subject-computer correlations for each verse are also shown.

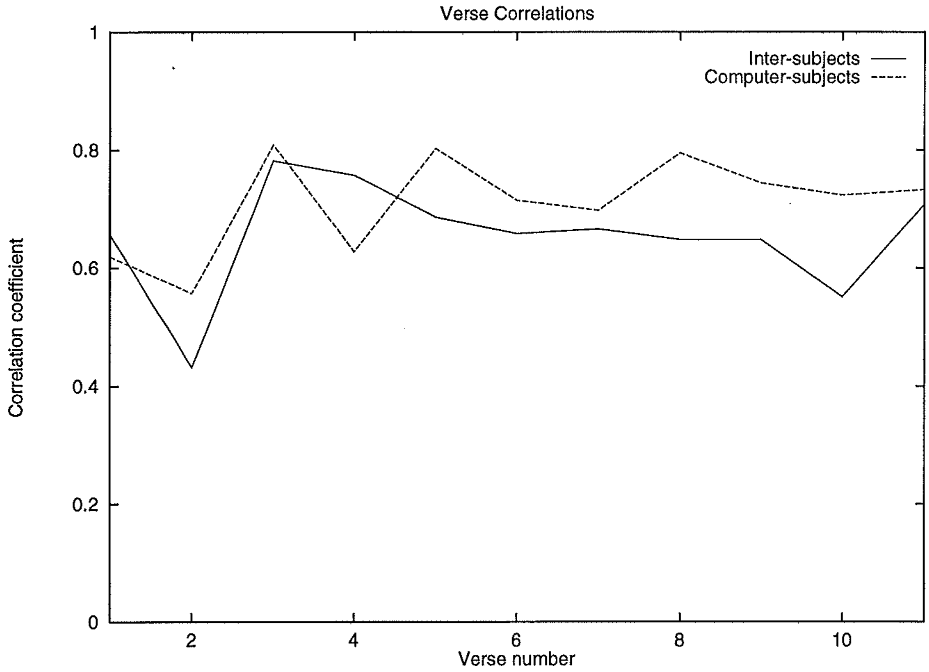


Figure 6.1: Computer-subject and Inter-subject Correlations per verse

As can be seen, there are some differences in rankings between subjects, but as semantic similarities are subjective, this is to be expected. However, the average inter-subject correlation is 0.65, indicating a reasonable consensus. The high values of computer-subject correlation suggest that the computer generated rankings lie somewhere near the average human-generated rankings. This is reflected in the following two graphs. Figure 6.2 shows the distribution of the raw human-computer rankings, where intensity is proportional to the number of occurrences. We can see that although there are outlying data-points where the computer and subjects disagree, the clustering of ranks shows a linear relationship.

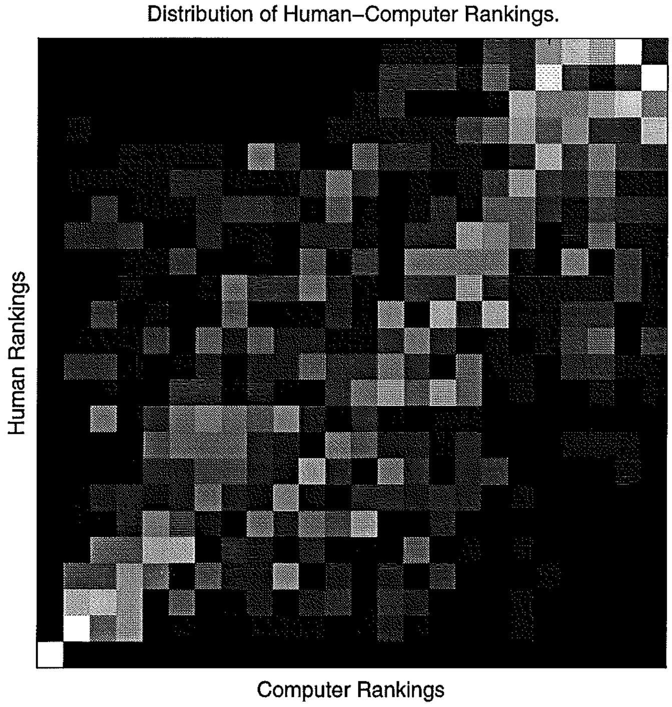


Figure 6.2: Distribution of Computer-Human verse rankings.

Another fact alluded to in Figure 6.2 but not reflected by the correlation coefficients is that the differences between human and computer similarities vary as a function of rankings. Figure 6.3 shows the average deviation for each rank number, and is generated from the 44 samples per ranking. The formula used to generate this graph is given by

$$d(R) = \sqrt{\sum_{s}^{subjects} \sum_{v}^{verses} (r_{s,v} - R)^2}, \tag{6.1}$$

where R is a ranking between 1 and 24, and s and v are taken from one of the four subjects and eleven reference verses. This graph shows that the computer-generated rankings reflect the human ratings more accurately at the extremities.

One reason for this is due to the structure of the Bible. Although there are many sections in the Bible that deal with a common topic, at the verse level these parallels are less frequent. As most verses in the Bible have less than ten conceptually related matches, a twenty verse similarity queue typically contains some less relevant associations. Accordingly, these verses had a wider range of similarity rankings, and hence a larger variance. In general, the four unrelated

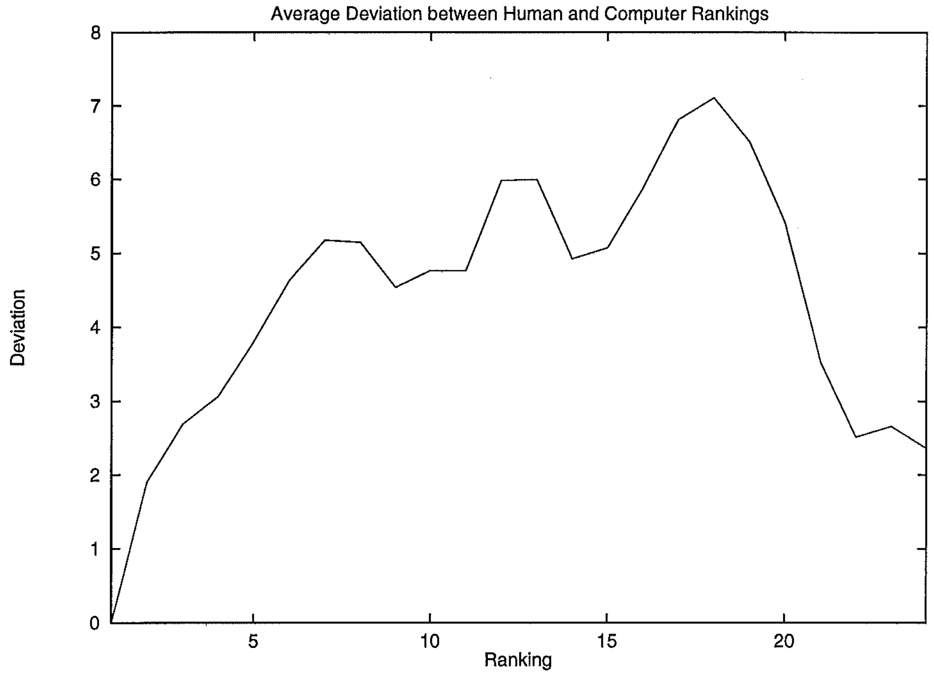


Figure 6.3: Average Deviation between Human and Computer rankings.

verses added to these lists have very few words or topics in common, and were thus consistently rated poorly by the computer and subjects alike.

The linear trend shown in Figure 6.2 can be emphasised by examining the average correspondence between human and computer-assigned rankings. The graph in Figure 6.4 was generated by averaging the human rankings across all subjects and verses for each of the 24 computer assigned ranks. While these averaged rankings belie the variation between the subjects and the computer, it still shows an obvious relationship between the two similarity rankings.

6.2.2 Sensitivity to Parameter Variations

All results described thus far apply to a statistical model with an inertia of $\rho = 0.4$ whose activation is spread over four iterations. The performance of the system was investigated as these two parameters were varied. To prevent new verses from being identified as the inertia and number of iterations are changed, these tests involved constructing a new search corpora for each of the lists.

Within each of these corpora, changes in similarity rankings were tracked, and the computer-subject correlations were then examined. This procedure allowed variations in rankings to be examined, but makes the assumption that these 24 verses would remain in the best-match lists.

While a conclusive study would generate new sets of similar verses for each different parameter value, this would require the volunteers to rank another 264

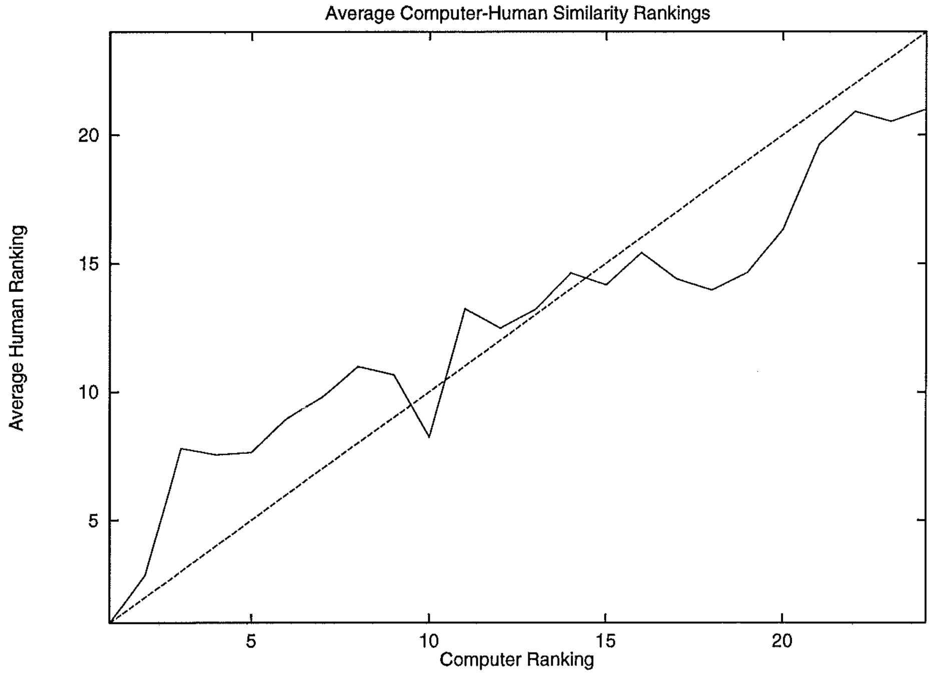


Figure 6.4: Average Computer-Human Similarity Rankings

verses for each new test. As this would correspond to over well over 3000 verses, this approach was abandoned, and the simpler approach taken.

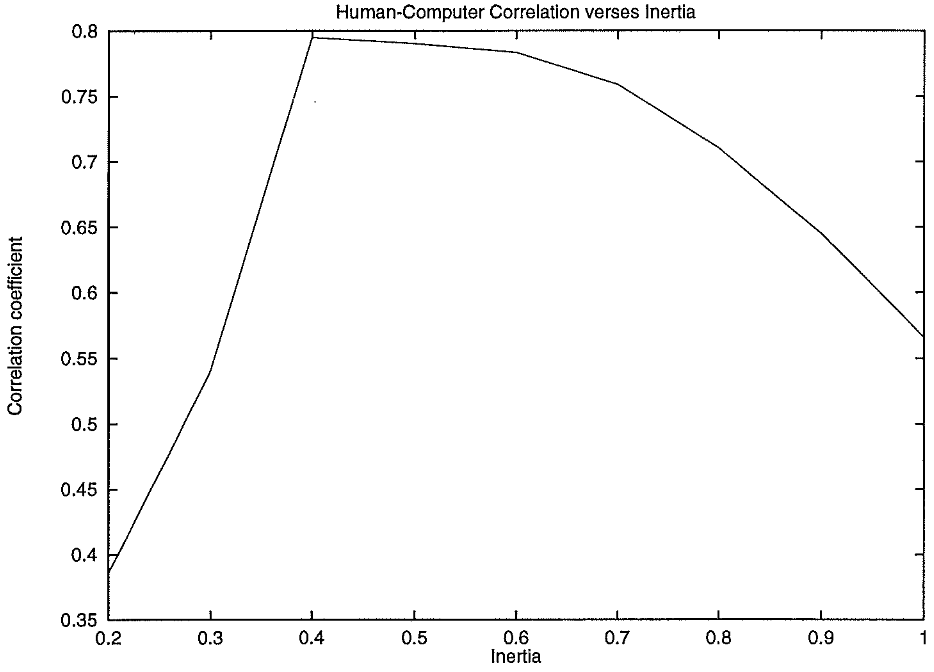


Figure 6.5: Computer-subject Correlation as a function of Inertia

Firstly, the dependence of the computer-subject correlation upon the inertia was investigated, the results of which are shown in Figure 6.5. As can be seen, the inertia was increased in steps of 0.1 from $\rho = 0.2$ to $\rho = 1.0$, where the searches for $\rho = 1.0$ correspond to a traditional keyword search.

For inertias less than 0.4 the program's ability to identify correct associations decreased rapidly; most likely a consequence of the energy being spread quickly from the search words to the extremities of the network. These results corresponded to the performance variations described in the self-verification experiment.

While the correlation was maximised when $\rho = 0.4$, we can see that the simple keyword search still performed reasonably well. However, from these results, and the two failures described in the self-verification experiment, we see that the statistical analysis of text can provide an obvious improvement in identifying similarities.

The other variation investigated was the system's dependence upon the number of iterations. When the energy was not propagated (corresponding to zero iterations), this search also corresponded to a keyword search. The variation in correlation as a function of iterations is shown in Figure 6.6.

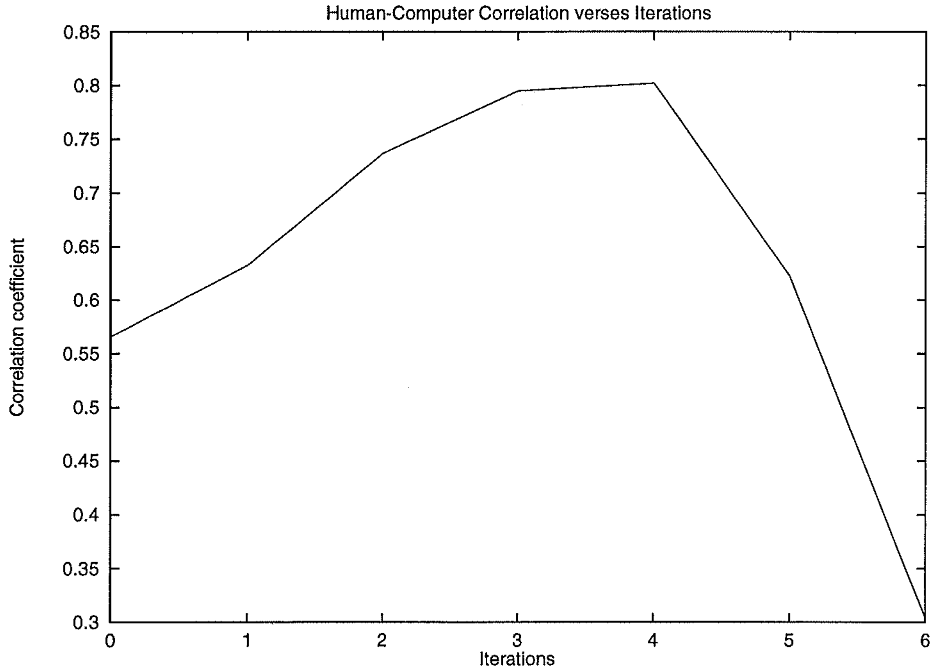


Figure 6.6: Computer-subject Correlation as a function of Iterations

As with the self-verification experiment, the system's ability to correctly rank similarities was maximised at four iterations, although the performance for less iterations was still acceptable. For more than four iterations, a marked decrease in performance is obvious.

One reason for the statistical text analyser's poor performance for larger number of iterations could be the network's high degree of connectivity. As mentioned earlier, the average number of links a node has is over 500, and so the energy from any one node can be spread throughout the entire network very rapidly.

For most search verses the entire network had been activated after three iterations, although most nodes had an negligible amount of energy. As the number of iterations is increased, however, the energy becomes more evenly distributed across the entire network, causing the model's entropy to increase, and resulting in the random selection of verses.

6.3 Conceptual Searches

The two preceding experiments involved the comparison between complete verses. While the program's performance has been shown to extend past that of a simple keyword search, there was still an overlap in words between matched verses.

In this section, the examples given were generated by specifying a search string which either had one or no words in common. This shows the ability of the program to rank the correct match in the top 20 of 31,000 verses based solely on conceptual similarity.

The results shown below were generated using an inertia of $\rho = 0.4$ over four iterations. Each example shows the search words used, with any word occurring in the matched verse being *italicised.* For each match found, the verse's ranking out of 31,000 is shown in **bold**.

1. cry, walls, valley, conquer, tumbled, *blast*

   (15) So the people shouted when the priests blew with the trumpets: and it came to pass when the people heard the sound of the trumpet and the people shouted with a great shout, that the wall fell down flat, so that the people went up into the city, every man straight before him, and they took the city.

   (18) And it shall come to pass, that when they make a long *blast* with the ram's horn, and when ye hear the sound of the trumpet, all the people shall shout with a great shout; and the wall of the city shall fall down flat, and the people shall ascend up every man straight before him.

2. bird, bill, plant, flood, *noah*, dried, return

   (8) And the dove came in to him in the evening; and, lo, in her mouth was an olive leaf pluckt off: so *Noah* knew that the waters were abated from off the earth.

3. bird, *ark*, flood, dried, return

(11) But the dove found no rest for the sole of her foot, and she returned unto him into the ark, for the waters were on the face of the whole earth: then he put forth his hand, and took her, and pulled her in unto him into the *ark*.

4. babylon threw prophet lion cave *god* saved faithfully

   (3) Then the king commanded, and they brought Daniel, and cast him into the den of lions. Now the king spake and said unto Daniel, Thy *God* whom thou servest continually, he will deliver thee.

5. *veil*, torn, half, tabernacle

   (8) And the *veil* of the temple was rent in twain from the top to the bottom.

   (14) And, behold, the *veil* of the temple was rent in twain from the top to the bottom; and the earth did quake, and the rocks rent;

From the above examples, we can see that even when a search phrase has only one word in common, the system is still able to correctly identify related verses. Each of the five searches ranked the desired verse in the top twenty, with the best match being ranked third overall.

6.4 Discussion

We have seen that the use of a statistical language model to identify semantic similarities offers a significant improvement over a simple keyword search. In the self verification tests, the spread of activation over the knowledge base provided the best search results, even when there were differences in the verses' vocabulary.

The results described in Section 6.2 have also shown that the measure of similarity used in this project approximates the human perception of semantic similarities to an extent. This was reflected by the nearly-linear relationship between the computer generated rankings and the average human rankings. Across all verses and subjects, the computer-human correlation coefficient was found to be 0.704.

Even when there is very little overlap between search phrase and the desired matches, the program was still able to correctly identify their semantic relationship. This shows that the use of spreading activation provides an effective way to identify related concepts within both the knowledge base, and the document being searched.

While the system's performance can be severely degraded under certain parameter values, near the optimal values it exhibits a degree of stability, and actual performance maximas may depend upon the type of texts being searched.

CHAPTER 7

Conclusion

The use of probabilistic language models for meaning analysis is still a largely unexplored area of natural language processing. This project has investigated the ability of these models to compare the underlying meaning present in natural language documents.

By using mutual information in a probabilistic spreading activation-based concept exploration system, I have shown that it is indeed possible to identify semantic similarities within a document. Moreover, my program was able to identify similarities from a corpus of over 31,000 verses. Within the top twenty computer-generated matches, the correlation between human and computer-generated rankings was found to exhibit a moderate linear relationship.

7.1 Contributions to Statistical Language Processing

These results contribute to the field of natural language processing in the following ways:

1. They show that the relationships identified by a modified n-gram language model can be used to identify semantic relationships. This differs from the traditional n-gram model which have been used to identify commonly occurring collocations and phrases.

2. They show that mutual information can be used as a measure of semantic similarity. Its suitability was examined both qualitatively, by assessing the relevance of the associations it generates, and quantitatively by using it to identify parallel passages within a corpus. The performance of the system when weighted using mutual information was found to provide an improvement over the traditional keyword search.

3. They show that spreading activation can be successfully applied to a probabilistic network. Traditionally, spreading activation has been applied to semantic networks. Even though some systems used incorporated a weighted

semantic network, no cases encountered have investigated its application to a statistically generated network.

4. I have implemented a statistical text-analysis program. Its features include the ability to construct a statistical knowledge base from a training corpus, to perform concept exploration across this knowledge base, and to search through a large corpus and identify the semantically related phrases contained therein.

5. I have demonstrated that a probabilistic language model can be used to provide a measure of semantic similarity. While some research had examined the semantic clustering of words based upon co-occurrences, no work was found that expanded this to concept identification.

7.2 Further Work

The work commenced in this project can be extended to address the following issues:

1. Would comparisons made on a larger passage of text provide a more meaningful method of identifying similarities within a document? Currently, my program compares semantic relationships on a verse level. As most topics in the Bible span several verses, it would be interesting to investigate how the system performs on larger topical units.

2. Can part-of-speech tags be used to improve the associations made by the program? Previous research has shown that the identification of tags can be used to distinguish between different syntactic instances of the same word, the size of the training corpus would have to be increased to provide a suitable number of training patterns.

3. How could the knowledge base be expanded to allow comparisons between different vocabularies? Given the existence of multiple versions of the Bible which preserve the verse divisions found in the King James Version, an examination of the training phase may allow the knowledge base to identify equivalent structures between different vocabularies. This would have implications for cross-language searches, without the need for machine-translation.

Although there has been little prior work in the field of statistical similarity analysis, the research described in this thesis has presented one possible way of using a statistical language model to identify semantic similarities. However, given the lack of previous research, it was not possible to investigate the improvements some proposed additions may have on the system.

References

1. Steven Abney. Statistical methods and linguistics. In Judith Klavans and Philip Resnik, editors, *The Balancing Act.* MIT Press, Cambridge, 1996.

2. J. R. Anderson. *Cognitive Psychology and its Implications.* W. H. Freeman and Company, San Francisco, 1980.

3. J. R. Anderson. *The Architecture of Cognition.* Harvard University Press, Cambridge, 1983.

4. A Averbuch, L Bahl, R Bakis, P Brown, A Cole, G Daggett, S Das, K Davies, S Gennaro, P de Souza, E Epstein, D Fraleigh, F Jelinek, J Moorhead, B Lewis, R Mercer, A Nadas, D Nahamoo, M Picheny, G Schichman, P Spinelli, D Van Compernolle, and H Wilkens. Experiments with the tangora 20,000 word speech recognizer. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, Texas, 1987.

5. Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque. Krypton: A functional approach to knowledge representation. In Wendy Lehnert, editor, *Strategies for Natural Language Processing*, chapter 24. Lawrence Erlbaum Associates, 1982.

6. Eric Brill. *Transformation-Based Part of Speech Tagger for Linux.* Center for Language and Speech Processing, John Hopkins University, Available at `http://www.cs.jhu.edu/~brill/home.html`, 1995.

7. Eric Brill, David Magerman, Mitchell Marcus, and Beatrice Santorini. Deducing linguistic structure from the statistics of large corpora. *DARPA Speech and Natural Language Workshop*, pages 380–389, 1990.

8. Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

9. Eugene Charniak. *Statistical Language Learning.* MIT Press, 1993.

10. H Chen, K Basu, and T Ng. An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic branch-and-boun search vs. connectionist hopfield net activation. Technical report, MIS Department, The University of Arizona, July 1994.

11. Hsinchun Chen, Bruce Schatz, Joanne Martinez, and Tobun Dorbin Ng. Generating a domain-specific thesaurus automatically: And experiment on FlyBase. Technical report, MIS Department, The University of Arizona, July 1994.

12. Paul R. Cohen and Rick Kjeldsen. Information retrieval by constrained spreading activation in sematic networks. *Information Processing and Management*, 23(4):255–268, 1987.

13. Cycorp. *The Cyc Technology.* Available at `http://www.cyc.com`, 1996.

14. Randall Davis and Bruce G. Buchanan. Meta-level knowledge: Overview and applications. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, chapter 22. Morgan Kaufmann Publishers, 1985.

15. R.G. Gallager. *Information Theory and Reliable Communication.* John Wiley & Sons, Inc., 1968.

16. Judy Kavanagh. The text analyzer: A tool for extranting knowledge from text. Master's thesis, University of Ottawa, 1995.

17. Allen Klinger. Natural language, linguistic processing, and speech understanding: Recent research and future goals. Technical report, Defense Advanced Research Projects Agency, 1973.

18. Wendy Lehnert, editor. *Strategies for Natural Language Processing*, chapter 1. Lawrence Erlbaum Associates Publishers, 1982.

19. Marvin Minksy. A framework for representing knowledge. In J. Haugeland, editor, *Mind Design*, pages 95–128. MIT Press Cambridge, 1981.

20. Paola Velardi and Maria Teresa Pazienza dn Michela Fasolo. How to encode semantic knowledge: A method for meaning representation and computer-aided acquisition. *Computational Linguistics*, 17(2):153–170, 1991.

21. Ralph Weischedel, Richard Schwartz, Jeff Palmucci, Marie Meteer, and Lance Ramshaw. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2):359–382, 1992.

APPENDIX A

Stop Words

The following words are defined as stop-words, and are ignored by the program during all processing stages. They include stop-words used by Kavanagh [16] and words from the corpus with very large frequency counts.

with
come
and
to
from
when
in
for
it
where
i
you
your
shalt
nor
three
six
nine
get
as
also
second
fifth
eighth
twentieth
fiftieth
eightieth
them
the
that
was
into
a
thou
which
be
have
an
if
its
itself
one
four
seven
ten
than
so
are
third
sixth
nineth
thirtieth
sixtieth
ninetieth
his
of
they
is
unto
not
thy
shall
by
ye
say
do
neither
goeth
two
five
eight
doth
doth
there
first
fourth
seventh
tenth
fortieth
seventieth
hundredth

twenty
thirty
forty
fifty
sixty
seventy
eighty
ninety
hundred
let
yet
any
no
art
also
some
us
am
at
then
o
or

APPENDIX B

Examples of Computer Matched Verses

Selected search verses used in the Computer-Human comparisons are shown below. Due to the size of the total test set, only three associations are shown for each reference verse.

1. I am the living bread which came down from heaven: if any man eat of this bread, he shall live for ever: and the bread that I will give is my flesh, which I will give for the life of the world.
    - This is that bread which came down from heaven: not as your fathers did eat manna, and are dead: he that eateth of this bread shall live for ever.
    - This is the bread which cometh down from heaven, that a man may eat thereof, and not die.
    - The Jews therefore strove among themselves, saying, How can this man give us his flesh to eat?
2. He that believeth on me, as the scripture hath said, out of his belly shall flow rivers of living water.
    - Verily, verily, I say unto you, He that believeth on me hath everlasting life.
    - Jesus cried and said, He that believeth on me, believeth not on me, but on him that sent me.
    - Verily, verily, I say unto you, He that heareth my word, and believeth on him that sent me, hath everlasting life, and shall not come into condemnation; but is passed from death unto life.
3. For God so loved the world, that he gave his only begotten Son, that whosoever believeth in him should not perish, but have everlasting life.
    - That whosoever believeth in him should not perish, but have eternal life.

- He that believeth on him is not condemned: but he that believeth not is condemned already, because he hath not believed in the name of the only begotten Son of God.
- In this was manifested the love of God toward us, because that God sent his only begotten Son into the world, that we might live through him

4. And he saith unto him, Verily, verily, I say unto you, Hereafter ye shall see heaven open, and the angels of God ascending and descending upon the Son of man.

    - And he dreamed, and behold a ladder set up on the earth, and the top of it reached to heaven: and behold the angels of God ascending and descending on it.
    - And John bare record, saying, I saw the Spirit descending from heaven like a dove, and it abode upon him.
    - And Jesus, when he was baptized, went up straightway out of the water: and, lo, the heavens were opened upon him, and he saw the Spirit of God descending like a dove, and lighting upon him:

APPENDIX C

Original Honours Proposal

Title:	A Statistical Approach to Text Similarity
Author:	Peter Dreisiger
Supervisors:	Dr. Mike Kalish (Psychology) Dr. Paul Hadingham (Computer Science)
CR Categories:	I.2.6, I.2.7

Aim

This research aims to identify similarities in meaning between parsed texts by comparing their surface forms. The study will investigate the possibility of reducing ambiguities in the input through the use of textual preprocessing and a probabilistic knowledge base whose state is dynamically updated to reflect newly parsed texts. The reasons for choosing a probabilistic model are twofold: firstly, in cases where a limited vocabulary can no longer be assumed, probabilistic models still offer the means of predicting a text's most likely interpretation [21], something that becomes infeasible for rule based parsers; and secondly, the learning ability of statistically based language models make them more robust when working with open-ended text.

The aim of the project is to produce a piece of software that can quantify the semantic similarities within a given corpus, and thereby identify associations between portions of text. While it is intended to compare these results with an existing program, it is hoped that this study will also examine how accurately the program reflects human experimental results. This set of manual comparisons may then be used as a reference when attempting to increase the percentage of correct associations the program generates.

Background

Since its inception in the early 1950's, artificial natural language processing's ultimate goal of endowing computers with an understanding of written text has gone largely unfulfilled. However, much research is still conducted in a field whose potential applications range from natural language man-machine interfaces and machine translators to complex literature search engines and meaning extractors.

The use of rule-based parsing and hand-annotated knowledge bases was first introduced during the 1970's [18]. While their use and the use of frames in some expert systems have been successful in dealing with restricted domains, the ability to correctly interpret the semantics of open-ended texts has been far more elusive.

Many attempts have been made at parsing natural language texts although in most systems, the requirements of a hand-crafted knowledge base become prohibitive once the domain restrictions are removed. Even so, Brachman et al. [5] found that by identifying terms and assertions, functional approaches could be used to implement a knowledge base. Conceptual distances have been used by Brachman et al. and Weischedel et al. [21] to provide a measure of association between objects in the knowledge base.

More recently, the use of statistical models in knowledge representations have been examined, and it has been found that a hybrid knowledge-based approach supplemented by a corpus-based probabilistic model can reduce interpretation ambiguity [21]. One advantage of incorporating a statistical model is its inherent ability to learn new words and update their associations. This feature can considerably reduce the amount of hand-crafted knowledge and definitions otherwise required. Another significant advantage a probabilistic model has over the use of formal logic is its tolerance to grammatical inconsistencies. This is partly because statistical methods compare surface models of the text being analysed while rule-based parsers often compare their representations; thereby restricting the input texts to ones that can be accurately represented.

One method used by Kavanagh [16] to identify concepts within a corpus involves the statistical analysis of word cluster densities although the relative occurrence of such objects within a sentence or paragraph has not been investigated. In the research by Kavanagh and Weischedel et al, it has been found that the use of textual preprocessing, and in particular part-of-speech tagging, can improve the identification of compound terms such as names and phrases. The availability of off-the-shelf preprocessors such as Brill's Part-of-speech Tagger [6], makes their incorporation into any new package significantly easier.

Method

Initially, I intend to read the available literature related to probabilistic knowledge representation schemes, techniques for the disambiguation of text and their use in relation to meaning extractors. In conjunction to this literature search, I will conduct a search for relevant programs and source code that may be usable as a basis for my project. Although ongoing, I expect this stage to take between 6–8 weeks.

The development will involve the modification or implementation of a statistical knowledge base and the subsequent training using a combination of readily downloadable corpora. The first stage will involve the construction of a semantics engine to analyse individual pieces of text, after which a method of evaluating the similarity can be investigated. I estimate this stage will take between 8–10 weeks. During this time, I intend to conduct an experiment under Dr. Kalish's guidance to identify the similarities between selected parts of texts as perceived by a group of test subjects.

The preprocessors will then be linked in and their effect compared against the reference program, my initial program and the manual associations derived in the experiment. This stage should take around two weeks. I anticipate the remaining 6–8 weeks will be spent collating results and producing the final dissertation.

This project will be implemented on a UNIX-based workstation, although the programming language used will be decided upon after further research.