University of Cincinnati

Date: 6/23/2015

I, Madhavun Candadai Vasu , hereby submit this original work as part of the requirements for the degree of Master of Science in Electrical Engineering.

It is entitled:

ANSWER : A Cognitively-Inspired System for the Unsupervised Detection of Semantically Salient Words in Texts

Student's name: **Madhavun Candadai Vasu**

This work and its defense approved by:

Committee chair: Ali Minai, Ph.D.

Committee member: Raj Bhatnagar, Ph.D.

Committee member: Carla Purdy, Ph.D.

UNIVERSITY OF Cincinnati

17314

ANSWER: A Cognitively-Inspired System for the Unsupervised Detection of Semantically Salient Words in Texts

A thesis submitted to the

Graduate School

of the University of Cincinnati

in partial fulfillment of the

requirements for the degree of

Master of Science (M.S.)

in the Department of Electrical & Computer Engineering

of the College of Engineering & Applied Science

by

Madhavun Candadai Vasu

B.Tech, Amrita School of Engineering, 2011

August 7, 2015

Committee Chair:	Ali A. Minai, Ph.D.
Committee Members:	Carla Purdy, Ph.D
	Raj Bhatnagar, Ph.D.

Abstract

Discovering salient words in text corpora is an important and largely unsolved problem in automated text analysis. This thesis describes a cognitively grounded, completely unsupervised, neurodynamical tool called *Attractor Network-based Salient Word Extraction Rule* (ANSWER) to accomplish this task. ANSWER is based on the hypothesis that salient words are disproportionately likely to occur in specific and coherent ideas, while non-salient words are likely to not show this bias. The core of ANSWER is a semantic network built from the pre-processed corpus such that each node is a neural unit denoting a particular word and the strength of the edges represent associations between those words based on information in the corpus. Attractor dynamics in the network causes the activity to converge to groups of strongly connected nodes, and these are seen as representing emergent *ideas*. Such ideas are sampled from the attractor network and salient word lists are drawn based on a score that computes the idea membership of a word. This thesis considers three distinct types of associative weights between words: Correlation coefficient, pointwise mutual information and joint probabilities. The effect of this choice on system performance is evaluated experimentally. The dependence of system performance on the main parameters is also studied and the system is found to be quite robust.

The most important features of ANSWER are that it is unsupervised and does not require that the corpus be divided into documents. This makes it scalable and broadly applicable to a wide range of corpora, unlike methods such as *Term Frequency - Inverse Document Frequency* (TF-IDF) that require that the text corpus be organized into distinct documents. In order to investigate ANSWER's applicability to different kinds of corpora, we applied it to three different kinds of corpora - a collection of technical abstracts from the proceedings

of a neural networks conference, a technical book and an autobiography. Performance is shown to be consistently good on all these corpora.

Saliency detection as a natural language processing task is a very well known problem and several network based and non-network based techniques have been proposed previously. This thesis compares ANSWER's performance with other standard saliency detection methods such as node degree, word frequency, betweenness centrality, eigenvector centrality, node weights, Max TF-IDF and Mean TF-IDF. ANSWER performs at least as well as the other methods and usually better.

While, ANSWER has been proposed as a tool for saliency detection, its underlying principles of idea generation make it suitable for extension to other natural language processing tasks such as named entity extraction, topic detection, document clustering, classification etc. Being a cognitively grounded model, ANSWER can also provide insight into other cognitive functions such as creativity and ideation.

Acknowledgement

I express my sincere thanks to my advisor Professor Ali A. Minai for all the support, guidance and understanding throughout my studying. He has been a great inspiration during my course of study and for my future plans. I would like to thank my committee members Dr. Carla Purdy and Dr. Raj Bhatnagar for reviewing my thesis and being part of my defense. I would also like to extend my thanks to Department of Electrical and Computer Engineering, University of Cincinnati for giving me an opportunity to go for graduate studies. Last but not least, my deep gratitude is to my family and my friends for all the love and support that I needed in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Cognition is the process of acquiring and organizing knowledge through experience. Understanding cognitive processes in necessary to explain behavior, perception and consequent action. Several researchers have come up with neurocomputational models that represent a specific cognitive function such as language [**?**, **?**], memory [**?**], motor control [**?**] etc. These models have been applied in machine learning paradigms where learning in machines is inspired from learning in humans or other animals. Typical applications include robot control [**?**], complex task learning [**?**] and natural language processing [**?**, **?**].

The work presented in this thesis focuses on modeling the cognitive processes related to natural language in terms of the dynamics of the underlying networks of neurons. Language is ambiguous and complex . Words have different meaning in different contexts. They could be used merely as a consequence of adherence to grammatical rules or they could be a key component in providing meaning to the text. Or a group of words could make collective sense given a specific context. Understanding natural language statements involves handling all these complexities effortlessly.

A key element in understanding linguistic expressions such as texts is the identification of salient words, i.e., words that carry important semantic content, as distinguished from words that serve only a grammatical or syntactic function. The method described in this thesis – called the *Attractor Network-based Salient Word Extraction Rule* (ANSWER) – is based on the principle that salient words are cognitively important, and are disproportionately likely to be used in concrete ideas. Thus, a fundamental requirement for the application of this principle to real texts is to extract ideas from it. To do this, a text corpus is seen in terms of associations between words occurring within a single semantic element, e.g., a sentence, and ideas are defined as combinations of mutually strongly associated words. This associative structure is instantiated as a recurrent neural network whose dynamics generates ideas as attractors, and can thus be seen as simulating the thought process [**?**, **?**, **?**]. Salient words are then extracted from these ideas by looking at their occurrence statistics.

The ANSWER approach has several advantages over previously proposed approaches to word saliency detection, including the following:

1. It is completely unsupervised and more practical than supervised methods.

2. Unlike methods based on document-wise word frequency, it can be applied to any sufficiently large corpus, regardless of whether it can be divided into documents. For example, ANSWER can analyze a single long text such as a book.

3. It extracts a comprehensive list of salient words rather than a small set of keywords.

4. It is rooted in cognitive mechanisms rather than statistical methods that are often counter intuitive and provide little insight.

1.2 Approach

The model is made up of an associative recurrent neural network where each neural unit denotes a word in the corpus. The words are connected to each other with weights based on their association in the corpus. Since associations are assumed to be symmetric, the network is an example of an attractor network [**?**]. When allowed to relax from an initial condition, the pattern of activity in the network converges to an attractor, where a small number of highly connected words are active and the rest are inactive. This set of active words is seen as representing an idea. Multiple ideas can be extracted either by repeatedly probing the network with different initial conditions – the approach used in ANSWER – or by allowing the current attractor to destabilize and freeing up the system to move to another attractor [**?**, **?**, **?**]. Details of the system are given in later chapters.

The setting up of this network is akin to the reading of the corpus and knowing its contents in a human mind. These semantic relationships create the network and the activity in the network is analogous to the thought process of exploring the semantic space set up by the corpus and coming up with ideas from it. Essentially, the creation of the network embeds in it a large repertoire of ideas as attractors, but these remain implicit until unmasked by the dynamics. The sampling of the system's state space with a few thousand initial conditions allows these latent attractors to be sampled, thus providing a collection of ideas that would be thought of by a mind after exposure to the corpus. Words that occur with disproportionate frequency in this collection of ideas are thus likelier to be salient than words selected from the original corpus by some simple criterion such as frequency of occurrence. This method can be seen as a twist on the widely used term frequency-inverse document frequency (TF-IDF) approach, where words that occur disproportionately often in a few documents are considered more salient. However, in ANSWER, the comparison is made between ideas and the corpus rather than between individual documents and the corpus as a whole, which is

more cognitively meaningful.

The performance on this task is compared against other widely used machine learning algorithms. The model is tested on three corpora of different types: abstracts of the Proceedings of the International Joint Conference on Neural Networks (IJCNN) ,a scientific book (Origin of Species by Darwin) and an autobiography (Playing it my way by Sachin Tendulkar).

1.3 Thesis Organization

The remainder of the thesis is organized as follows:

Chapter 2 reviews relevant research, gives basic definitions and provides some notes on the tools that were used in implementation.

Chapter 3 elaborates the concepts behind the design, provides the mathematical formulation and describes how it was adapted for the saliency detection task.

Chapter 4 presents results, compares the performance of the system with other techniques, and includes discussion on the results.

Chapter 5 concludes the thesis with a summary of the results and analyses. Several important directions for future research in this area are also suggested.

Chapter 2

Background and Relevance

This chapter cites other work in similar areas and compares our model with reference to them. Broadly, the proposed model falls into the category of neural models based on associative learning and recall. While this model has been applied to detect salient words from a corpus, it is also a model of semantic cognition via the emergent phenomenon of idea generation based on word associations. Previous work, both from our lab and other research groups based on these two perspectives are discussed below. The first section primarily discusses this thesis in light of previous work on theoretical and experimental aspects of cognitive semantics. The next section focuses on computational methods of natural language processing, especially salient words detection.

2.1 Semantic Cognition

It has long been recognized that associations between semantic elements, e.g., concepts, categories, etc., is fundamental to many cognitive processes – especially those involving language and ideas. Semantic cognition primarily attempts to answer questions such as - how a word is attached to its meaning? How does a group of words presented as a sentence in a

specific order convey the intended meaning? Conceptual semantics studies the representation of meaning and understanding and lexical semantics studies word relations and contextual meaning.

In order to understand how individuals associate words with their meanings and how words with similar meanings are connected in the mind, word association experiments have been performed extensively. Typically, subjects are given a word and they are asked to provide another word representing the same or similar concept in response[**?**, **?**]. These experiments have resulted in the creation of huge databases that provide relative strengths of associations between words[**?**] based on subject responses. These studies however, have not gone beyond the bigram (pairs of words) or sometimes trigram (word triplets) associations. As the number of words increase the associations tend to become less and less generic and the group of words tend to denote a specific concept.

While these studies allowed free associations, some studies on recall have shown that probability of recalling a list of random words decreased when cued with associated words that were supposedly meant to assist recall [**?**, **?**]. In other words, associations created a bias in the search and recall for words from semantic memory. Recall based on a cue was shown to be dependent on the pre-existing associations with the cue. These theories suggest that a cue causes the words associated with the cue to be activated and hence only help recall of associated words while it diminishes the chances of recalling non-associated words[**?**, **?**]. Spreading activation theories [**?**] support the idea that associative networks represent these search and recall processes effectively. These studies show that a word prime causes a spread of activity across a range of associated elements, represented as nodes in the network, and the activity eventually reaches the target. This phenomenon simulates the search and consequent recall of associated elements. These studies have also shown that the effect of the prime can be subdued by stimuli that intervene the effective spreading of the activation[**?**, **?**]. For a word prime, the target could either be a word or any associated semantic element. An

attractor network has proven to be a good model to represent the process of priming followed by spreading activation to reach a target [?] and theories backed by experiments have shown that both internal and external stimuli could alter the effect of the prime. The stimuli were theorized to affect network states and cause network state transitions that may or may not lead to the target associated with the prime, depending on the nature of the stimulus [?].

So far, we have seen that associations play a key role in memory and recall and that attractor network theories with spreading activation confirm to experimental evidence of priming. These theories bode well to the studies that formulate ideation as an associative process[?]. Creativity was understood complex as a process that thrives on associations and to be greatly affected by situational attributes[?, ?]. There have been many studies that theorize the mental processes of intuition, incubation, serendipity ,the entrepreneurial and scientific ingenuity and so on [?, ?]. Studies have analyzed specific individuals [?], specific scenarios such as brainstorming sessions[?]. Creative ability in science has been studied as a constrained process that combines the ability to retain selectively and sustain focus on a limited space [?, ?].

Language exhibits all the above characteristics - two words that mean same thing or represent the same concept are associated; sentences are made up of words that together convey a specific meaning. Thus a language can be represented as a network in order to study its structure and evolution [?]. Several studies have been conducted along these lines. One such large study lead to the development of Wordnet, a lexicon that provides a huge list of words, their meanings and semantic relationships[?]. It consists of about 155,000 nouns, verbs, adjectives, and adverbs. It also includes simplex words like put, phrasal verbs like put up, and idioms like put out the dog. This network has been studied [?] and found to have small world characteristics and so did the thesaurus based associative network study by Motter [?] and studies on other semantic networks[?]. Studies of other languages have yielded similar results [?]. Such graph based networks of words have been constructed for

individuals based on free association experiments and the structure of those networks have been analyzed [**?**]. Creativity has been studied as a function of structure of these semantic networks. Individual with a broader, more diffuse associative network have been identified as more creative than individuals with few selective associations.

2.2 Natural Language Processing (NLP)

Machine processing of natural language is a very extensively researched area. Electronic text is available from several sources in extremely large quantities. There is a pressing need to process unstructured data to glean insights and extract information. One of the most fundamental problems in automatically making sense of unstructured text is identifying salient words. The main challenge here lies in identifying the salient words based on context because based on context the meaning, role and importance of a word changes significantly in the English language.

The first step in NLP is to restructure and represent the data in an organized fashion. Network models have been used extensively to represent text corpora for text processing. Nodes in the network could represent a document in the corpus or words in the corpus and the edges could mean several different kinds of associations. Naturally, such networks can be a very effective way to visualize text corpora. First, logical form triplets (Subject-Verb-Object) are automatically extracted and their named entities are created as head nodes. Second, verbs are created as child nodes of these head nodes followed by the rest of the extracted elements as their child nodes. This creates a semantic graph of the document [**?**] and provides a very easy to understand visualization to unstructured text [**?**]. Furthermore, Support Vector machines have been used on these semantic graphs to extract sub-graphs that can be compiled to create a summary[**?**]. These techniques have shown to create summaries that are of similar quality to human made summaries. Semantic graphs have also been used

to look up answers to questions posed to a system. The natural language question is parsed to obtain named entities and subject-verb-object triplicates and the semantic graph is looked up based on these details. Upon identifying a similar elements on the graph the answer is returned along with the document that it is contained in[**?**].

Another method of creating semantic graphs are known as Distance Graphs [**?**] where each node denotes a word in the corpus. The edges are defined based on the occurrence of those words within an interval of at least k of each other. The parameter k is known as the order of the graph. The distance graph preserves some amount order information in the corpus because these are directed graphs. These have been applied to identify similar documents by comparing the graphs and in plagiarism detection by matching sub-graphs. Undirected versions of the distance graph has also been proposed for flexibility to adapt them for other languages where word ordering may not act the same way as in the English language.

From all the above work it is apparent that network models of text are very effective in representing text corpora and have wide variety of application in the field of natural language processing. The specific problem of interest in this work is detecting salient words in a corpus. There have been several approaches to this problem, many of them not based on network models. Different approaches to salient word detection follows.

The frequency based approach proposed by Luhn[**?**], was perhaps the first automated approach to detecting salient words. Words with very high or very low frequency were discarded as non-salient and words that were in between were regarded salient. Abstracts were created automatically by putting together sentences based on the words in them. Salient sentences were found based on the frequency of words in the sentences and from them, automated abstracts were prepared. A more effective and sophisticated approach was proposed by Salton and colleagues [**?**, **?**]. This method is known as Term Frequency - Inverse Document Frequency method (TF-IDF). It calculates the saliency of word v in a document d

as:

$$s(v|d) = TF(v,d) \times IDF(v) = \frac{n(v,d)}{N_d} \times log\frac{M}{m(v)} \tag{2.1}$$

where $n(v,d)$ is the number of times v occurs in d, N_d is the number of words in d, M is the total number of documents in the corpus, and $m(v)$ is the number of documents that contain the word v. This measure assigns a relatively higher score for a word that occurs frequently in d but does not occur in most documents. Words that occur with a high frequency over all documents in the corpus receive a relatively low score. Thus, this method of salient word detection identifies salient words at the level of each document which can be combined by averaging over documents or by any other means to arrive at a corpus level TF-IDF measure. The words can then be ranked according to this measure and high ranking words can be deemed salient. The major drawback of this method is that it requires the corpus to be organized into distinct documents.

Adding another degree of complexity to Luhn's frequency based approach, Ortuno[**?**] analyzed recurrence interval of words. The standard deviation of the distance between successive occurrences was shown to be a great metric for automatic keyword identification. The Bible was processed and keywords were presented. Recurrence intervals have also been analyzed with respect to a stochastic process model. F-statistic measures have been used to distinguish keywords from noise words[**?**]. Another statistical measure that has been used is Shanon's entropy of information. This metric captures the amount of information in a word using Shanon's entropy then ranks words and retrieves keywords [**?**].

Computing the inhomogeneity of a word's local versus global density based on the premise that salient words have a "bursty" distribution, occurring with greater density in local neighbourhoods where they are more relevant rather than uniformly over the corpus [**?**] i.e. in the spatial distribution of words in a document, salient words tend to be present as clus-

ters rather than spread across the entire document. While it might seem that this method will fail when there are several documents because salient words may be present in several clusters when we look at spatial distribution of words in the merged corpus, it has been shown that such a merger would cause the metric to increase value uniformly across all words and hence still allow detection of salient words. Another similar approach based on word occurrence and distribution is based on analyzing a word's distribution relative to its neighbors distribution in the corpus [**?**]. In this approach, single and multi-word concepts are extracted based on distributions of individual words and groups of words respectively. A metric is arrived at for these n-tuples based on the distribution of the n words in the corpus and their combined presence in the corpus. this score is used as a metric to extract single word concepts - salient words.

All the above approaches to salient word extraction were not network based. Grinva[**?**] has proposed a network based measure. A semantic network is constructed with each word in the corpus as a node and edges denoting the semantic relatedness between the terms as calculated by the links between those terms in the Wikipedia library. A community extraction algorithm was employed over the network. A criteria that identified groups containing salient words was presented and the other groups were discarded.

The above ideologies and theories set up the stage for designing a semantic cognitive model for representing natural language. The proposed model is based on a neurodynamical model of thinking called *IDEA* (*itinerant dynamics with emergent attractors*) that has been described previously [**?**, **?**, **?**, **?**] by other members of our lab, and earlier work on computational models of ideation and priming [**?**, **?**, **?**, **?**]. In short, the proposed model is a recurrent associative network where the nodes represent words and are modeled as neural units. Activity spreads across the network based on word associations and attractors embedded in the network are made up of groups of well associated words. Choosing to use a network model would solve some of the drawbacks of certain other techniques such as the need for

corpus to be organized as documents (TF-IDF). Also, constructing the semantic graph from the corpus would make the system unsupervised as opposed other systems that required a training phase, hence making this system more practical in application. While this network has been applied to detecting salient words in this case, this network could potentially be seen as a model for intuitive understanding of language and idea generation. With that perspective in mind, this network could be applied to other cognitive processes by suitably defining the nodes and the edges.

2.3 Tools

The free and open source Natural Language Toolkit (NLTK) python package from *http://www.nltk.org/* provides an easy-to-use interface for handling unstructured text data. This package provides in built functions that stop, stem and manipulate text data. A different toolkit has been adopted by another member of the lab to perform the stopping and stemming procedures. However, the processing following those steps, such as tokenizing, finding frequency distributions of words etc. has been performed using NLTK.

Another member of the lab, Mei Mei used Adobe Acrobat to convert PDF files of the text corpora to Html files. She processed them using JAVA code and converted the Html files to text using *"html2txt"* that is available for free from *http://www.nirsoft.net/utils/htmlastext.html*.

The NumPy python package which is also a free and open source package from *http://www.numpy.org/* provides the MATLAB like matrix manipulation and scientific computation functions. The setting up and dynamics of the associative neural network are implemented using the functions in this package.

Gephi, a free and open source network visualization and analysis tool available at *http://gephi.github.io/* was used to compute network measures such as Eigenvector Centrality, Betweenness Centrality etc.

Eclipse IDE was used to work with the python development and MATLAB was used to analyze the results and generate plots.

Chapter 3

System Description

This chapter provides the technical description of ANSWER. Details regarding the intuition behind the design, the steps involved in preprocessing the corpus, the statistical information obtained from the corpus that helps set up the network, the mathematical formulation for the dynamics of the network, and its use for determining lexical saliency are all discussed.

3.1 Design Philosophy

As discussed in Chapter 1, the main difficulty in extracting salient words from text is that saliency is primarily a *semantic* attribute that is only imperfectly represented by lexical statistics. In particular, any extensive text ends up using many words for functional reasons – i.e., reasons of grammar, syntax, rhetorical convention, etc. – which adds a lot of "lexical noise" from a semantic viewpoint, and makes it difficult to distinguish the semantically salient words by their frequency or some such simple statistical metric. In corpora with distinct documents, this difficulty can sometimes be overcome by using methods such as TF-IDF, but this is more difficult to apply in corpora comprising a single long document, such as books.

One alternative is to apply the tools of natural language processing to the problem, with syntactic parsing, grammatical analysis and the use of an appropriate ontology. This is useful in well-defined domains, but scales poorly for general corpora – especially as they grow in size. The main motivation behind the ANSWER approach is to try and capture some of the semantic information needed for detecting salience through purely statistical means. ANSWER does this by drawing upon the mechanisms of cognition through a neural network model. This approach is based on three essential postulates:

- *Semantic knowledge is fundamentally associative*: Semantic knowledge derived from a text corpus is represented in the mind by an *Associative Semantic Network* (ASN) whose nodes are words (or concepts) and where the edges between nodes represent associations between pairs of words as found in the corpus. The ASN built based on these statistics obtained from the corpus can thus be seen as a semantic network formed in the mind as a result of reading the corpus.

- *Ideas are defined by associative coherence*: Ideas are formed by the association of mutually harmonious words, i.e., words that "go together" based on the patterns the mind has learned from experience. Thus, in an associative network of words, ideas correspond to small groups of words with strong mutual associations among all the words.

- *Ideas concentrate salience*: Semantically salient words are disproportionately likely to occur in coherent ideas, whereas semantically non-salient but frequent words are likely to occur at or below their overall frequencies. In particular, relatively low frequency words in the corpus are likely to be much more frequent in ideas if they are semantically salient and vice versa.

Of course, these postulates represent a heuristic and intuitive rather than a formal argument. The work in this thesis is an attempt to validate them through application. However,

it is best to see ANSWER as part of a larger toolbox for analyzing salience rather than a universal tool. In particular, it is useful to apply the standard method of removing words whose non-salience is clear by other means, e.g., stop words such as articles, pronouns, prepositions, etc., and apply ANSWER only as a final tool to decide among the "difficult" words that the simpler tools have not been able to remove.

The main requirement for the application of the approach outlined above is to discover ideas from arbitrary corpora. There are several potential approaches to this, but here we use an approach based on a neurodynamical computational model of thinking called the *Itinerant Dynamics with Emergent Attractors* (IDEA) model [**?**, **?**, **?**]. In this model, ideas a defined as emergent attractors in an associative lexical neural network with competitive dynamics. However, the attractors are not explicitly encoded into the network, but become embedded implicitly as the associative connectivity specified by experience. Thus, the embedded attractors have to be extracted by a *sampling process* as described later in this chapter.

One important difference between the IDEA model and ANSWER is that the latter does not use itinerant attractors generated in a sequence. Rather, the ANSWER system is sampled repeatedly from random initial conditions and allowed to converge to an attractor each time. This leads to a set of independent rather than correlated samples, conditioned on the weights of the network. The methodology of sampling the attractor space and extracting salient words is explained in later sections.

3.2 Corpora

The experiments in this thesis use three corpora - one comprising a set of conference paper abstracts, and the other two representing books. Each of these is described here:

1. **The IJCNN Corpus**: Abstracts from the 2009, 2011 and 2013 proceedings of the International Joint Conference on Neural Networks (IJCNN). This corpus consists of

1410 abstracts that can be considered as independent documents. Since the corpus is organized into independent documents, document specific measures such as TF-IDF can be employed. Although the documents are independent, since all abstracts deal with the same general area of Neural Networks "lexical noise" is likely to be distributed similarly across all documents. A hand labelled list of salient words was created for this corpus and the performance of ANSWER and other methods were compared against this list.

2. **"Origin of Species" by Charles Darwin**: This is a technical book presenting Darwin's theory of evolution. Unlike the IJCNN corpus, this book is only a single document and does not have distinct sub-units and as a result measures such as TF-IDF cannot be computed on this corpus. This book was chosen because in order to test ANSWER on a single unit of large text and also to diversify the nature of corpora we tested ANSWER with.

3. **"Playing It My Way", by Sachin Tendulkar**: This autobiography of the cricket player, Sachin Tendulkar was chosen because unlike IJCNN it is a non-technical book and is bound to contain a very different use of the language and unlike "Origin of Species" it is a contemporary book that would contain a current slang and vocabulary. Here, also, the entire book is treated as a single document and hence measures such as TF-IDF cannot be applied.

3.3 Pre-processing

Text corpora are typically obtained in their raw formats from various sources. Files are usually PDF or plain text (txt) files. These documents need to be preprocessed to prepare them for further analysis. In the case of PDF files, extra processing is required to parse

the file and extract just the content of interest. Figures, tables, headers, footnotes etc. need to be removed and unnecessary formatting information needs to be deleted. Since the associative network in ANSWER is constructed based on the sentence as basic unit, the corpus is split into its constituent sentences. Pre-processing involves preparing the corpus with these initial levels of processing followed by extracting word tokens, calculating their frequencies, and preparing a matrix of counts for the number of times each pair of words occur in the same sentence. Statistical information that is encoded in the weights between neural units is computed from this data.

All steps explained in this section involving pre-processing a text corpus, C, were done by another member of the lab, Mei Mei. She used Adobe Acrobat to convert PDF files to Html files. Html files provide font information so that irrelevant content can be filtered out easily. She then used the "html2txt" tool to convert all processed html files to text files *http://www.nirsoft.net/utils/htmlastext.html* . Stemming and stop word removal were the performed as follows:

1. A Porter stemmer that is available at *http://www.cs.cmu.edu/∼callan/Teaching/porter.c* was used to stem the words. Words in the set that stemmed to the same root were replaced by a single reference word from the set to make all words in the final dataset recognizable (e.g., "use", "user", "using", "uses" were all replaced by "using").

2. Standard stop words were removed using the list at: *http://norm.al/2009/04/14/list-of-english-stop-words/.*

3. A heuristic algorithm described in [**?**] was used to remove further non-salient words. A *relative prominence* value, $R(v_i)$, was calculated for each word, v_i, using two quantities: $f_{ELP}(v_i)$, its frequency (in occurrences per million words) in the 40,481-word *English Lexicon Project* (ELP) corpus (elexicon.wustl.edu); and $f_C(v_i)$, its frequency in the corpus under analysis, C. The relative prominence was given by:

$$R(v_i) = log \frac{f_C(v_i)}{f_{ELP}(v_i)} \tag{3.1}$$

All words v_i with $R(v_i) < 0.001$ were removed. The logic behind this step is that salient words are likely to occur at higher frequencies relative to their frequencies in the English language in general. The threshold of 0.001 was determined through trial-and-error and fixed for all corpora.

At the end of the pre-processing, only relatively salient words remain. ANSWER's task, therefore, is to filter this list further as a final step. The final processed corpus, C_p, is made up of N_S sentences, N_W word tokens, and N_V unique words. The vocabulary of unique words is denoted by $V = \{v_i\}$.

3.3.1 Weight metrics

Following the first level pre-processing explained above, the next step is to obtain statistics from the data to build the associative network. Every sentence in the corpus is represented as a set of word tokens without repetition, i.e., each unique word in the sentence receives only one token regardless of how many times it occurs in the sentence. Statistics are computed based on the number of sentences that have a particular word, and the number of sentences that have each pair of words. In order to reduce the size of the data and the strong but unwarranted effect of the low-frequency words, words that occur fewer than 4 times (i.e., in fewer than 4 sentences) in the corpus are eliminated.

The *occurrence probability*, p_i, of each word, v_i, is given by the fraction of sentences that include the word, and the *co-occurrence probabilities*, p_{ij}, for every pair of words, v_i and v_j, are given by the fraction of sentences that include both words. These basic statistics can then be used to calculate several possible associative metrics between words, such as conditional probabilities, correlations, etc. A priori, it is not clear which of these metrics is most suitable

for detecting semantic salience. Most of the metrics have cognitively plausible associative meanings, and several can be inferred by reasonable neurobiological learning processes. One of the most important aspects of the work in this thesis is to compare three of the most promising associative metrics empirically over several text corpora, and to provide guidelines on their use.

Each of the three metrics studied captures the strength of association between pairs of words based on their sentence-wise co-occurrence in the corpus. In particular, the associative value a_{ij}, between neural units i and j represents the association between words v_i and v_j. All three metrics are symmetric, leading to a symmetric association matrix, which guarantees the existence of attractors [**?**]. The three metrics used are as follows:

Joint Probability: This metric considers two words to be more highly associated if they co-occur in a larger fraction of sentences. This is the most obvious measure of association, and is given directly by the co-occurrence probabilities calculated during data processing:

$$a_{ij} = p_{ij} \tag{3.2}$$

The main expected drawback of using this metric is that higher frequency words tend to acquire larger associative values with many other words, and thus tend to dominate in the estimation of salience.

Correlation Coefficient: This metric considers two words to be associated if they have a higher correlation coefficient based on their individual and joint probabilities. If two words v_i and v_j co-occur more frequently than implied by chance based on their individual probabilities, then the words are considered more associated:

$$a_{ij} = \frac{p_{ij} - p_i p_j}{\sqrt{p_i(1-p_i)}\sqrt{p_j(1-p_j)}} \tag{3.3}$$

This metric has the advantage that it quantifies statistical dependence, i.e., the amount

of association not explained by pure chance. It is conveniently confined to the range between -1 and $+1$, and is also biologically plausible under reasonable scenarios of Hebbian learning [**?**]. The main drawback is that rare words can acquire high associative value if their few occurrences happen to be in the same sentence. This is mitigated somewhat by dropping very low frequency words. Another drawback of this metric is that it captures only linear, i..e., first-order dependence.

Pointwise Mutual Information (PMI): The PMI metric tries to approximate the full statistical dependence between words v_i and v_j using the idea of mutual information. It is defined by:

$$a_{ij} = log\frac{p_{ij}}{p_i p_j} = logp_{ij} - logp_i p_j \tag{3.4}$$

Like the correlation coefficient, this metric too captures the difference between the actual probability of co-occurrence, p_{ij}, and the probability expected if the co-occurrences were purely random, i.e., $p_i p_j$. However, this metric is bounded by $(-\infty, min(log1/p_i, log1/p_j)]$ rather than $[-1, +1]$, and can potentially be more informative than the correlation coefficient. PMI has been widely used as a measure of word association in computational linguistics following the work of Church and Hanks [**?**]. The main drawback with PMI is that it is quite sensitive to poor estimation of probabilities due to inadequate sampling, which can be the case with small corpora and/or rare words.

Before using these association values in the neural network, two further steps are done. First, all negative values are set to zero. This potentially loses some useful information about dissociation between words, but in practice, negative association values are quite small and are more of a nuisance than anything else in the neural dynamics. As will be described below, the network equations already incorporate inhibition between all neurons as part of the competitive dynamics, which makes the small negative associations less relevant. Second,

the weights in the network are normalized to the 0-1 range by dividing all weights by the maximum weight. This is done mainly to make the network weights comparable so that the same parameter values can be used with networks using each of the three weight types.

Networks with each of the three weight types were used with each of the corpora, and the results are described in the next chapter.

3.4 The Network Model

This section provides a brief description of the recurrent neural network model used in ANSWER.

3.4.1 Model Structure

The neural network is a one-layer recurrent network of $n = N_V$ neural units, where N_V is the number of distinct words in the corpus after pre-processing. Each neural unit corresponds to a unique word, and the connections between the units represent associations between the words. The network has competitive K-of-n dynamics, i.e., only the K or so most stimulated of the n neural units are allowed to be active at any given time, and thus potentially represent an idea with K words. K is a parameter of the system that controls the size of the ideas. Every unit is connected to every other unit with a weight that corresponds to joint probability, correlation coefficient, or PMI between the words that those units represent. Each neural unit computes the weighted sum of all inputs and generates a binary 0/1 output. There are no specific inhibitory influences between the units beyond the global lateral inhibition implied by the competitive activity rule, and the units do not influence themselves i.e. the weights are all greater than or equal to zero and the self weights are zero.

3.4.2 Model Dynamics

A stable state of the network is defined as a non-transient converged activity state, i.e., when all the units of the system reach steady-state values of 0 or 1. Such a state is termed an *attractor*. A random cue is given to the network by externally activating a subset of K neurons. After cueing, the network, it is allowed to act on its own until it converges to an attractor. The active neurons in the attractor are then read off as a group of words comprising an emergent idea.

The input received by unit i at time t is given by:

$$x_i(t) = \sum_{j=1}^{n} a_{ij}(t) z_j(t-1) + \gamma_{noise} \xi_i(t) \tag{3.5}$$

where z_j is the output of unit j, γ_{noise}, is a gain parameter, and $\xi_i(t)$ is uniform white noise. The state of unit i is updated at time t using:

$$y_i(t) = \alpha y_i(t-1) + \frac{(1-\alpha) x_i(t)}{\sum_{j=1}^{n} z_j(t-1)} \tag{3.6}$$

The value of α is set just below 1 to simulate continuous-time dynamics and the denominator is a normalizing factor. The K-of-n global inhibition rule works as follows. Based on a predefined value of K, the top K neurons that are currently the most excited are allowed to fire at that time step. This sets up a limit on the number of neurons that could win at any time step and corresponds implicitly to using inhibitory lateral weights between all neurons. The equation for the output of unit i is:

$$z_i(t) = f(y_i(t)) = \begin{cases} 1, & \text{if } i \in \{\theta\% \text{ of } K \text{ most excited units and } y_i(t) > 0\} \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

Based on past experience with the *IDEA* network, it has been determined that a rigid K-of-n rule leads to numerically-induced fluctuations, and so a soft version of this rule has been implemented where any unit, i, with $y_i(t)$ within $\theta\%$ of the nominal K-of-n threshold is also allowed to fire. Also, only neurons with positive excitation are allowed to fire. Thus, while a particular value of K is set as a parameter, the actual number of active neurons at any given time may be somewhat greater or lesser than K.

To summarize, activity in the system is initiated by activating a subset of the neural units, which causes the activity to spread across all associated units. At each small time step a soft thresholded K number of neurons emerge as the winner. The activity eventually converges to a state where a subset of the neurons will remain active forever unless the network is further disturbed. This set of neurons represent the emergent idea from the semantic network.

3.5 Saliency Detection

Once the network is built as described above based on the data from pre-processing the corpus, it represents the entire corpus and can be seen as embedding the ideas inherent in the corpus. These ideas are sampled through a large number of random cueing trials, and the resulting attractors are analyzed to determine word saliency.

3.5.1 Random Cueing

The experiment to retrieve a list of ideas from the corpus involves cueing the network repeatedly with random cues. In each trial, K units are chosen randomly and activated, with the remaining units set to zero. The network dynamics is then allowed to converge to an attractor. The *idea* that the activity converges to is noted. This process is repeated with 5,000 random cues for each trial and for 7 such independent runs.

In each trial, a saliency score for each word is computed based on the normalized idea membership of the word as follows

$$S(v_i) = \frac{\sum_{k=1}^{N_c} z_i^k}{N_c} \tag{3.8}$$

In the case of the IJCNN corpus, performance is analyzed for every trial and an average performance is computed over 7 trials. Whereas, in the other two corpora, since performance cannot be computed in every trial, salient words are extracted by averaging the saliency score for each word over 7 trials. More about this is explained in the next chapter.

where N_c is the total number of cues and z_i^k denotes the i^{th} bit of the binary attractor vector generated by the k^{th} cue. Words that have saliency scores over a set threshold are deemed salient. The number of words that are deemed salient depend strongly on the threshold. Hence the threshold can be seen as a tool to concentrate salience and also as a method to generate salient word lists of a desired length.

3.5.2 Parameter Settings

The primary parameter that controls the functioning of ANSWER is K, which decides the number of neurons that are allowed to be active at any time step. As explained before, this is only a soft threshold with a θ% margin. In terms of ideas, K and θ control the size of the ideas i.e. the number of words that form an idea. Naturally, we would not want ideas to be too large or very small, however we should also study the effect of this parameter on the performance of the system. To this effect, we have run the random cueing experiment with three different values of K = 5,7 and 10. Parameter θ is kept at a constant value of 2 and is not changed because it only acts as a fine adjustment over the value of K and does not significantly affect the size of the ideas. Other variables requiring setting are: α, which is set at a constant value of 0.9 to simulate a continuous time system, and γ, which controls

the influence of noise that is added at each time step is kept at a constant value of 0.1.

The other critical parameter in ANSWER is the frequency threshold, used to determine whether a word occurs with sufficient frequency in the attractors to be deemed salient. Changing this can make the saliency requirements more or less stringent, and work in this thesis systematically explores the effect of varying this threshold. The threshold values are expressed as a percentage of the normalized maximum frequency, and the values used in the experiments are 0.2%, 0.3%, 0.4%, 0.5%, 0.6%, 0.8%, 1%, 1.5%, and 2%. Based on the value of K and the thresholds, the random cueing experiment yields word lists of different lengths. These values have been chosen so as to get word lists of various tangible lengths, ranging from long lists that could potentially cover all salient words in the document to short word lists that capture just the keywords of the corpus. The ability to pick up salient words by the network at various thresholds is tested and results are presented in the next chapter.

3.6 Comparison with Other Methods for Saliency Detection

It is unrealistic to expect any purely statistical method to capture the semantic salience of words perfectly – especially since salience is itself hard to define precisely. As such, the utility of a method such as ANSWER lies mainly in its *relative advantage* over other methods of similar complexity. Thus, the reported studies compare ANSWER against several intuitive and widely-used statistical methods of estimating word salience. Since there is no objective measure of word saliency, the comparison between methods uses subjective but independent assignments of word saliency by human coders as the ground truth.

In order to provide a fair comparison, each of the other methods is applied to generate word lists of exactly the same size as those generated by ANSWER at each setting of K and the saliency threshold. The resulting lists are validated against the labelling provided by the

human coders.

The statistical saliency detection methods used to compare with ANSWER are the following:

1. *Frequency*: Word lists of the required length are obtained by first sorting the words in the corpus by frequency and selecting the most frequent words. Only the final vocabulary that remains after removing stop words is used in this procedure.

2. *Mean TF-IDF*: The TF-IDF value for each word is calculated over all documents using the formula shown in chapter 2. The mean of these values is used as the TF-IDF score for each word and the words are ranked based on this. Words with a greater mean TF-IDF value are ranked higher and word lists of required length are obtained by thresholding at this list size. This metric can be used only on corpora that are organized into distinct independent documents, e.g., the IJCNN corpus.

3. *Max TF-IDF*: This metric simply uses another approach to combining the TF-IDF scores for a word in each document. While the previous method computes the mean, in this method, words are ranked by the maximum TF-IDF value over all documents. This means that if a word is prominent even in one document, it is more likely that it is deemed salient because it will be ranked higher.

4. *Node Degree*: Considering the sentence as a semantic unit, a network has been constructed where words are connected to words that have co-occurred in the same sentence. In this case, words are ranked based on the number of edges incident on them, i.e., their nodal degree, with higher degree nodes being ranked higher. Thus, words used in conjunction with a large number of other words are deemed more salient. This method also requires that we remove stop words before constructing the network.

5. *Node Weight*: The nodal degree metric only counts the number of incident edges, but

disregards how strongly the words are associated with each other. In this case, words are ranked based on the summed association weights of their incident edges. Higher cumulative weight nodes are ranked higher and word lists of specific lengths are drawn based on the rank. Unlike the node degree case, here words that may have fewer but stronger weights might get picked up as salient.

6. *Betweenness Centrality*: The betweenness centrality of a node in a graph is defined as the fraction of shortest paths between all node pairs that pass through the node [**?**] . It is widely used in network analysis as a measure of node significance. Here, words are ranked such that nodes with higher values of betweenness centrality are regarded as more salient.

7. *Eigenvector Centrality*: This is a recursive network measure [**?**] where the significance of a node is measured based on the proportion of the strength of its connection with other significant nodes. Google uses a variation of this measure to rank web pages in their search results [**?**]. Here, word lists are prepared such that words having greater eigenvector centrality are ranked higher.

The network measures that are described above were calculated using *Gephi*. A network is defined in *Gephi* with each word being the node and the number of times two words co-occur in a sentence as the edges. The co-occurrence count is normalized to create the edge weights. *Gephi's* in built network analysis tools are run and the results are exported. TF-IDF is computed using the NLTK package in python.

3.7 Summary

To summarize, the text corpus under consideration is first parsed and read as sentences. The first level of pre-processing involves stemming and stop word removal. After omitting words

that have fewer than 4 tokens, 3 different kinds of statistics are obtained from the corpus, namely joint probabilities, correlation coefficients, and point wise mutual information. This information is used to set up the networks. The networks are randomly cued a large number of times and the emergent attractors are saved for each weight metric. Based on the emergent attractors, a saliency score is computed, words are ranked and word lists are obtained based on setting a threshold on the saliency score. Word lists of same length are obtained from the other standard statistical salient word extraction techniques and their performances are compared.

Details of the experiments and their results are given in the next chapter, along with a discussion of these results.

Chapter 4

Results and Discussion

As described in the previous chapter, ANSWER has been applied to three different text corpora - IJCNN, "Origin of Species" and "Playing it my way". One major obstacle in analyzing ANSWER's performance is the availability of ground-truth data to compare against. A definitive list of salient words in a corpus is virtually impossible to obtain, since there is typically no objective criterion for determining saliency. Thus, the performance of ANSWER has been evaluated in the following ways:

1. **Evaluation of System Dynamics:** ANSWER's dynamics, stability and efficiency have been analyzed for different types of associative weights: correlation coefficient, mutual information and joint probability.

2. **Comparison with Other Methods:** Using subjective designations of salient words by humans, the ability of ANSWER to discover these words has been compared with that of the other methods described in Chapter 3. The performance metric here is the density of salient words in the list returned at various saliency thresholds for a given weight type and K.

3. **Comparison of Weight Types:** Using the same process as in item 2 above, the results for the three types of associative weights have been compared at various values of the saliency threshold and K.

4. **Evaluating the Effect of Varying K:** The performance of the system has been compared for K =5,7 and 10. As discussed in the previous chapter, K is one of the most important parameters of the network. The values of K that were chosen here were selected so as to test the network for dependence of K, while at the same time keeping in mind that it denotes the size of ideas and ideas are comprised of a reasonably small number of words. The soft threshold parameter θ controls how elastic the size of the ideas can be.

Two different methods were employed to validate the performance

- In the case of the IJCNN corpus, Dr.Ali Minai manually annotated the vocabulary to designate the ground-truth baseline list of salient words in the corpus. This list has gone through several revisions and can be considered a fair list of salient words to compare against.

- In the case of "Origin of species" and "Playing it my way", estimating the performance in the absence of a ground truth validation list is carried out by having several individuals pick out false positives from the word lists each algorithm generates and compiling their responses to estimate efficiency. Lists of equal lengths made up of the top N salient words from each method were given to individuals who were familiar with the corpus. The lists were anonymized so that the participants would not know which algorithm created which list. Participants marked the words that they felt were non-salient in the context of the book and their responses were quantified using the following metrics -

1. *Minimum Rule* - A word was termed non-salient even if just one evaluator marked it as non-salient.
2. *Majority Rule* - A word was marked non-salient only if a majority of the evaluators identifies it as salient (at least 3 out of 5 evaluators).
3. *Maximum Rule* - A word was deemed non-salient only if all evaluators marked it as non-salient.

Performance in both these cases was computed as a percentage of true positives in the word list. While performance on the IJCNN corpus has been analyzed based on the ground truth list, results for "Origin of Species" and "Playing it my Way" have been presented using these metrics along with samples of word lists created by each algorithm. This chapter gives more insight into ANSWER's performance in each corpus and discusses the results from the perspectives outlined above.

4.1 IJCNN corpus

The IJCNN corpus consists of abstracts from the proceedings of the conference from 2009, 2011 and 2013. There were 1410 documents in all out of which 6 were rendered empty after pre-processing. The 1404 documents that remained were made up of N_S =12,011 sentences, N_W = 99,169 word tokens. Upon removing words with less that 4 tokens there were N_V = 2,309 unique words.

4.1.1 Results

The results shown below present the performance of ANSWER in comparison to other standard saliency measures. Setting a specific threshold and K results in ANSWER producing a word list of a certain length, N_{trial}. Since all the other measures are ranking based measures,

we can obtain word lists of the same length as obtained from ANSWER. The efficiency of each of these measures is computed as a percentage of the true positives in the respective word lists. This kind of quantitative analysis is possible only because of the presence of a "ground-truth" salient word list for this corpus.

$$Efficiency = \frac{number\ of\ salient\ words\ identified}{total\ number\ of\ words\ identified} \tag{4.1}$$

As discussed previously, networks with three different weight types – correlation coefficients, mutual information and joint probability – were used. The network dynamics was studied for $K = 5$,7 and 10. A comparison of the performance of ANSWER for all combinations of weight type and K versus the other standard methods is shown in Tables 4.1 through 4.9. All data has been obtained by averaging over 7 trials for each case with 5000 random cues in each trial. Tables 4.10 through 4.12 give the standard deviations of the performance over the seven trials of each case to indicate that the performance was extremely stable and repeatable. Tables 4.13 through 4.15 compare the performance of networks with different associative weight types. The manually marked list of salient words consisted a total of 1667 words, which means that a random designation of word saliency would yield a 72.2% efficiency. This is considered as the baseline performance.

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	547	381	277	212	139	100	41	15
ANSWER	80.46	81.45	82.40	83.12	82.74	81.64	85.65	99.10
Frequency	75.29	77.18	75.26	75.30	74.15	76.33	70.66	44.38
Degree	76.80	76.81	78.21	76.91	76.72	78.82	85.23	93.17
Weights	74.14	75.01	74.44	73.89	72.82	70.18	66.49	59.09
Mean TF-IDF	75.37	76.36	75.00	75.09	73.75	75.95	71.02	79.20
Max TF-IDF	70.38	72.02	69.94	70.84	71.66	72.15	64.69	63.27
Eigenvector Centrality	71.92	70.86	71.19	73.07	70.75	72.58	64.54	81.82
Betweenness Centrality	72.13	71.23	71.82	73.07	74.51	75.91	71.30	54.54

Table 4.1: Comparison of performance with different methods. Weight type: Correlation coefficient, K =5, $\theta = 1$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	907	666	503	388	240	162	68	25
ANSWER	81.26	82.37	84.09	84.56	85.49	84.42	87.22	97.27
Frequency	78.27	76.75	75.51	77.38	74.27	73.70	74.87	63.29
Degree	78.87	76.79	76.39	76.79	78.07	77.93	82.11	87.10
Weights	78.25	75.89	73.67	75.06	73.49	72.55	73.00	62.30
Mean TF-IDF	78.27	76.80	76.25	76.09	74.74	73.42	74.87	72.38
Max TF-IDF	71.72	70.98	70.91	72.15	69.69	72.98	68.62	59.97
Eigenvector Centrality	71.97	72.02	71.28	70.68	71.71	71.20	67.09	73.41
Betweenness Centrality	72.35	71.99	72.73	71.30	73.14	71.91	68.60	60.30

Table 4.2: Comparison of performance with different methods. Weight type: Correlation coefficient, K =7, $\theta = 1$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	1449	1176	945	759	514	361	163	74
ANSWER	78.67	79.97	81.21	82.57	84.09	84.91	86.55	92.61
Frequency	77.98	79.01	78.53	77.25	75.56	76.84	73.83	76.15
Degree	80.48	80.34	79.14	77.49	76.50	76.96	78.11	82.14
Weights	79.36	79.43	78.91	76.57	73.72	75.97	72.44	73.29
Mean TF-IDF	77.44	78.19	78.66	77.13	75.84	76.83	73.56	76.15
Max TF-IDF	71.69	71.59	71.75	71.40	70.73	71.81	72.71	69.15
Eigenvector Centrality	71.46	71.40	72.19	71.21	71.47	70.98	71.27	68.74
Betweenness Centrality	71.77	71.78	72.13	72.19	72.70	70.54	71.74	70.10

Table 4.3: Comparison of performance with different methods. Weight type: Correlation coefficient, K =10, $\theta = 1$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	687	477	352	243	128	61	21	13
ANSWER	82.05	85.41	85.53	86.29	83.85	85.90	87.06	89.68
Frequency	76.80	75.98	76.92	74.21	74.48	72.80	56.17	47.62
Degree	77.04	76.26	77.01	77.78	77.34	83.72	90.56	92.06
Weights	76.11	73.39	75.97	73.52	71.87	71.71	56.17	65.87
Mean TF-IDF	76.75	76.53	76.53	74.90	75.26	72.80	72.53	84.13
Max TF-IDF	71.16	70.95	71.80	69.82	72.66	68.50	61.56	60.32
Eigenvector Centrality	71.71	71.51	71.05	71.74	69.27	65.17	80.82	84.13
Betweenness Centrality	72.08	72.70	70.67	72.98	77.34	68.50	58.14	52.38

Table 4.4: Comparison of performance with different methods. Weight type: Mutual Information, K =5, $\theta = 1$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	845	641	501	382	227	138	49	22
ANSWER	80.91	82.84	84.42	85.16	86.06	85.01	84.25	86.04
Frequency	78.11	76.43	75.57	77.20	74.49	74.11	71.22	58.32
Degree	78.75	76.64	76.36	76.77	77.86	77.02	82.89	87.77
Weights	77.60	75.39	73.64	75.02	73.61	72.40	68.50	58.32
Mean TF-IDF	77.72	76.53	76.30	76.42	74.34	74.13	69.17	72.21
Max TF-IDF	71.92	71.08	70.97	72.14	70.10	71.91	69.17	60.20
Eigenvector Centrality	71.77	72.32	71.17	70.83	71.85	70.20	63.00	80.03
Betweenness Centrality	71.84	72.38	72.70	71.27	73.46	74.68	72.61	57.02

Table 4.5: Comparison of performance with different methods. Weight type: Mutual Information, K =7, $\theta = 1$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	1065	876	725	579	393	273	110	59
ANSWER	80.82	82.07	83.16	84.00	84.82	84.85	88.46	85.28
Frequency	78.57	78.19	76.72	75.37	77.52	75.18	76.90	72.32
Degree	79.82	78.76	77.09	76.81	76.93	78.37	75.99	84.19
Weights	79.22	77.96	76.31	74.62	74.98	74.32	69.61	71.17
Mean TF-IDF	78.72	78.00	76.77	75.31	76.00	75.18	73.26	72.32
Max TF-IDF	71.12	71.56	71.02	70.88	72.26	69.80	72.05	68.93
Eigenvector Centrality	72.12	71.60	71.02	72.04	70.74	71.52	72.34	64.39
Betweenness Centrality	72.25	72.29	72.31	71.98	71.41	72.27	75.68	68.93

Table 4.6: Comparison of performance with different methods. Weight type: Mutual Information, K =10, $\theta = 1$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	113	84	64	50	36	29	19	16
ANSWER	75.82	78.11	75.95	71.96	69.44	68.14	51.75	45.83
Frequency	76.71	76.10	73.71	71.33	72.22	65.79	51.75	43.75
Degree	76.09	80.88	82.82	84.01	88.89	86.32	94.64	93.75
Weights	69.90	71.75	72.13	68.59	66.67	62.37	53.61	56.25
Mean TF-IDF	73.44	78.46	73.71	70.64	72.22	72.79	75.05	75.00
Max TF-IDF	71.97	70.90	68.08	68.00	61.11	63.60	64.33	62.50
Eigenvector Centrality	71.70	69.70	65.81	62.68	63.89	68.26	83.92	81.25
Betweenness Centrality	76.09	73.67	68.08	71.43	69.44	64.72	57.12	56.25

Table 4.7: Comparison of performance with different methods. Weight type: Joint Probability, K =5, $\theta = 0.9$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	483	321	249	200	144	115	82	60
ANSWER	76.90	74.97	73.80	74.87	74.52	77.43	77.14	72.22
Frequency	75.80	76.43	74.20	74.70	74.07	76.83	76.34	72.22
Degree	76.34	77.47	77.94	76.37	76.62	76.49	81.24	83.90
Weights	73.45	75.39	73.93	73.88	72.91	70.12	72.26	71.65
Mean TF-IDF	76.62	75.70	74.47	74.70	73.62	73.60	77.94	72.78
Max TF-IDF	70.97	71.13	69.92	71.05	72.21	72.44	70.21	68.90
Eigenvector Centrality	71.52	70.82	71.93	73.03	71.31	71.33	69.78	64.97
Betweenness Centrality	72.69	70.82	72.73	72.21	73.16	76.21	73.03	68.88

Table 4.8: Comparison of performance with different methods. Weight type: Joint Probability, K =7, $\theta = 0.9$

Threshold	0.003	0.004	0.005	0.006	0.008	0.01	0.015	0.02
# of words	2301	2210	1854	1264	472	268	114	76
ANSWER	72.12	72.27	73.60	74.98	76.32	76.14	75.14	76.86
Frequency	75.80	72.86	75.46	78.77	76.07	74.99	76.61	76.42
Degree	76.34	89.10	77.87	80.40	76.49	78.49	76.31	82.10
Weights	73.45	74.10	77.81	79.40	73.52	74.25	70.17	72.93
Mean TF-IDF	76.62	72.67	75.19	78.11	76.70	75.00	73.68	76.42
Max TF-IDF	70.97	72.54	71.89	71.46	70.90	69.90	71.92	69.86
Eigenvector Centrality	72.29	72.50	72.52	71.72	71.54	71.65	71.65	69.86
Betweenness Centrality	72.28	72.62	72.23	71.49	72.60	72.65	76.31	71.17

Table 4.9: Comparison of performance with different methods. Weight type: Joint Probability, K =10, $\theta = 0.9$

Threshold	K = 5	K = 7	K = 10
0.003	0.640	0.285	0.299
0.004	0.834	0.496	0.275
0.005	1.322	0.530	0.464
0.006	0.682	0.561	0.498
0.008	0.936	0.983	0.860
.01	1.754	1.084	0.556
.015	5.899	3.885	2.438
.02	2.362	2.980	2.098

Table 4.10: Standard deviations of performance. Weight type: Correlation coefficient

Threshold	K = 5	K = 7	K = 10
0.003	1.553556	0.760221	0.613936
0.004	0.926035	0.72744	0.788707
0.005	0.353419	0.538449	0.117912
0.006	1.407265	1.035401	0.482531
0.008	0.909787	1.07498	0.405651
.01	1.165142	1.060186	0.821878
.015	2.89601	1.063929	1.224903
.02	3.436609	4.827976	1.387077

Table 4.11: Standard deviations of performance. Weight type: Mutual Information

Threshold	K = 5	K = 7	K = 10
0.003	1.043	0.099	0.327
0.004	0.871	0.182	0.118
0.005	1.832	0.403	1.306
0.006	2.801	0.863	0.124
0.008	2.778	1.776	0.712
.01	1.701	1.253	1.668
.015	1.519	0.662	0.230
.02	3.608	1.362	0.487

Table 4.12: Standard deviations of performance. Weight type: Joint Probability

Threshold	ANSWER-C		ANSWER-M		ANSWER-J	
	# of words	efficiency	# of words	efficiency	# of words	efficiency
0.003	547	80.46	709	81.38	113	75.82
0.004	381	81.46	479	84.34	84	78.11
0.005	277	82.40	350	85.71	64	75.95
0.006	212	83.13	240	87.92	50	71.96
0.008	139	82.75	129	84.50	36	69.44
0.01	100	81.64	61	86.89	29	68.14
0.015	41	85.66	20	90.00	19	51.75
0.02	15	99.11	12	91.67	16	45.83

Table 4.13: Comparison of performance for different weight types. K =5; ANSWER-C, ANSWER-M and ANSWER-J correspond to ANSWER set up with correlation coefficient, mutual information and joint probabilities respectively.

Threshold	ANSWER-C		ANSWER-M		ANSWER-J	
	# of words	efficiency	# of words	efficiency	# of words	efficiency
0.003	907	81.26	839	80.81	483	76.90
0.004	666	82.38	637	82.42	321	74.97
0.005	503	84.09	504	84.33	249	73.80
0.006	388	84.57	387	84.24	200	74.87
0.008	240	85.49	224	84.82	144	74.52
0.01	161	84.42	138	84.06	115	77.43
0.015	68	87.22	50	84.00	82	77.14
0.02	25	97.28	22	81.82	60	72.22

Table 4.14: Comparison of performance for different weight types. K =7; ANSWER-C, ANSWER-M and ANSWER-J same as above

Threshold	ANSWER-C		ANSWER-M		ANSWER-J	
	# of words	efficiency	# of words	efficiency	# of words	efficiency
0.003	1449	78.68	1070	81.21	2301	72.12
0.004	1176	79.97	877	82.10	2210	72.27
0.005	945	81.21	725	83.03	1854	73.60
0.006	759	82.58	572	83.57	1264	74.98
0.008	514	84.09	395	84.81	472	76.32
0.01	361	84.92	262	85.11	268	76.14
0.015	163	86.56	112	88.39	114	75.14
0.02	74	92.62	61	86.89	76	76.86

Table 4.15: Comparison of performance for different weight types. K =10; ANSWER-C, ANSWER-M and ANSWER-J same as table above

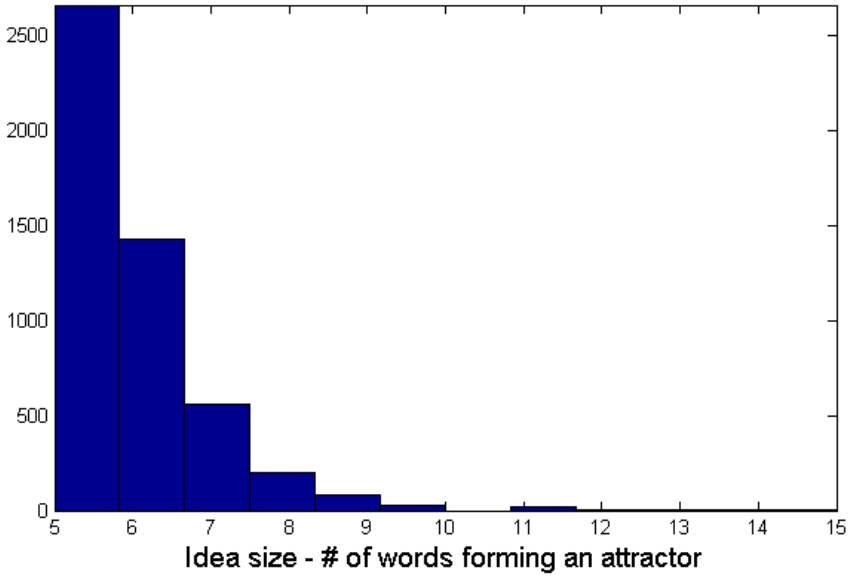

Figure 4.1: A histogram of the size of ideas over one trial of 5000 random cues. Weight type: Correlation coefficient, K =5

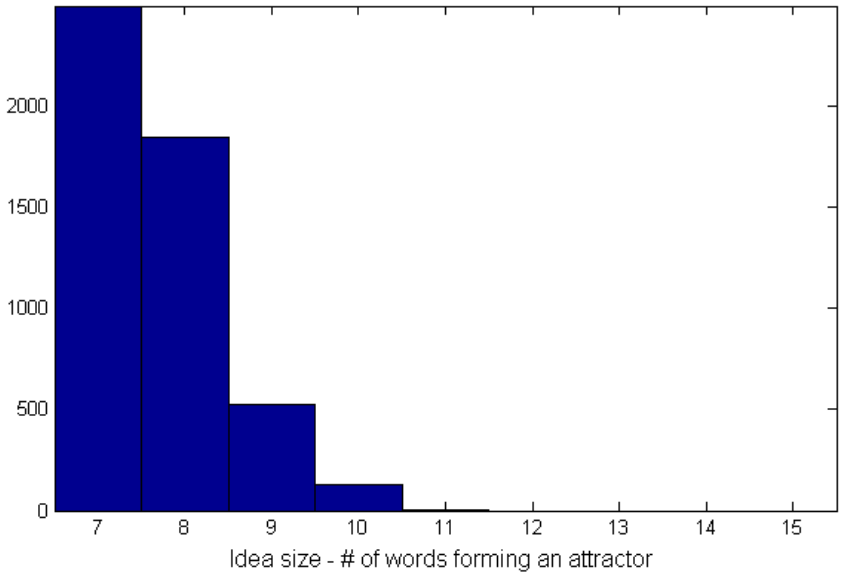

Figure 4.2: A histogram of the size of ideas over one trial of 5000 random cues. Weight type: Mutual Information, K =7

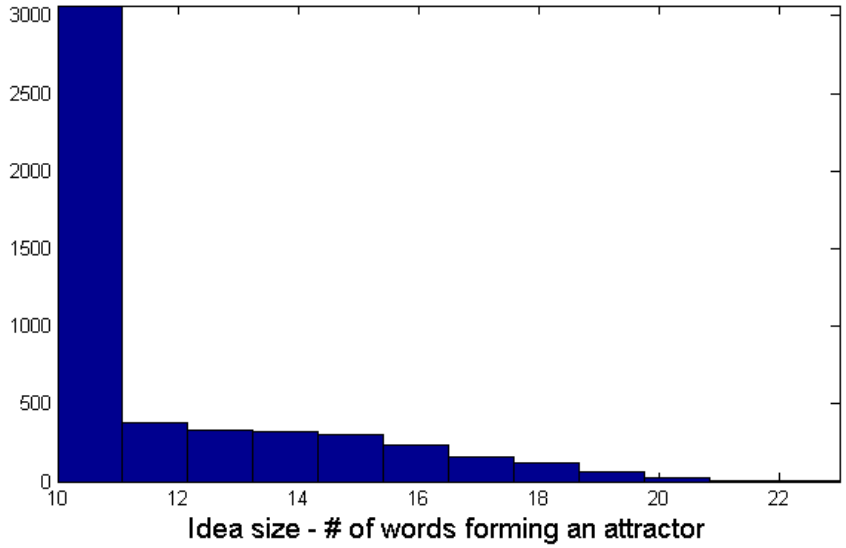

Figure 4.3: A histogram of the size of ideas over one trial of 5000 random cues. Weight type: Joint Probability, K =10

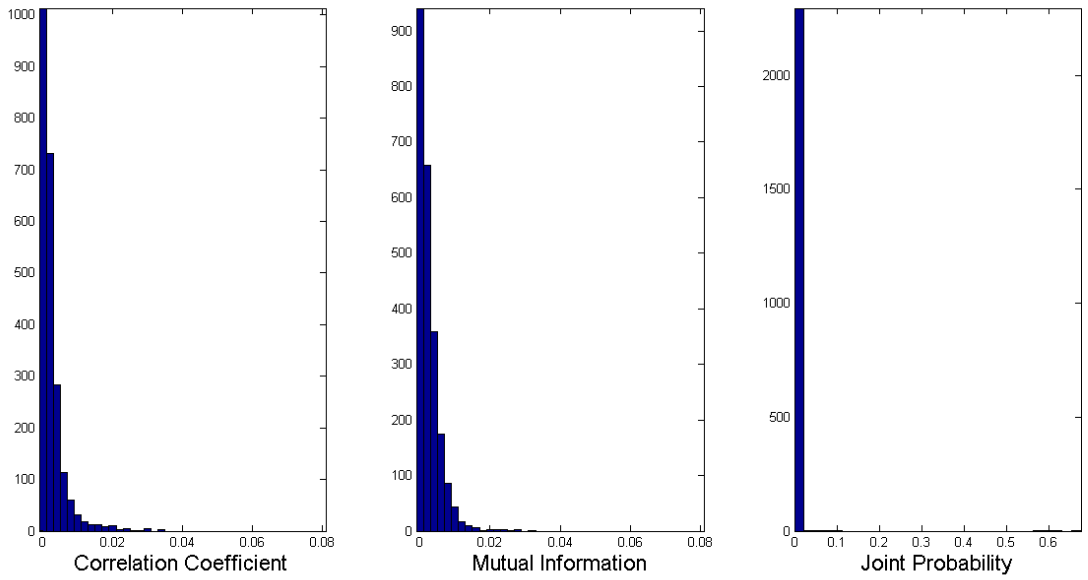

Figure 4.4: A histogram of the saliency score averaged over all trials. K =5

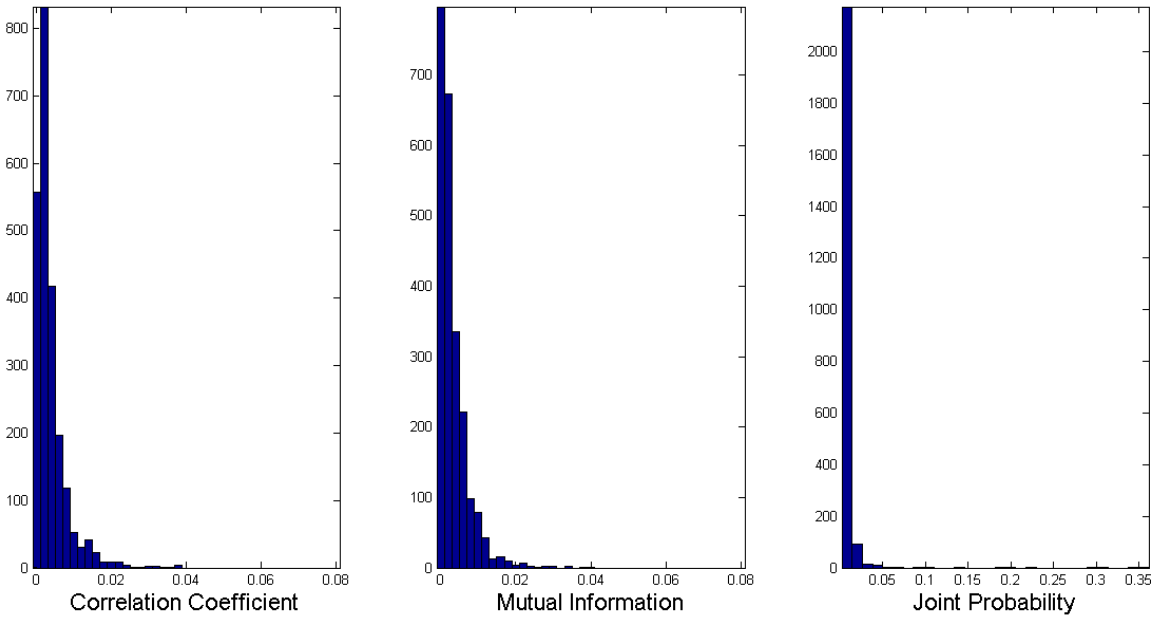

Figure 4.5: A histogram of the saliency score averaged over all trials. K =7

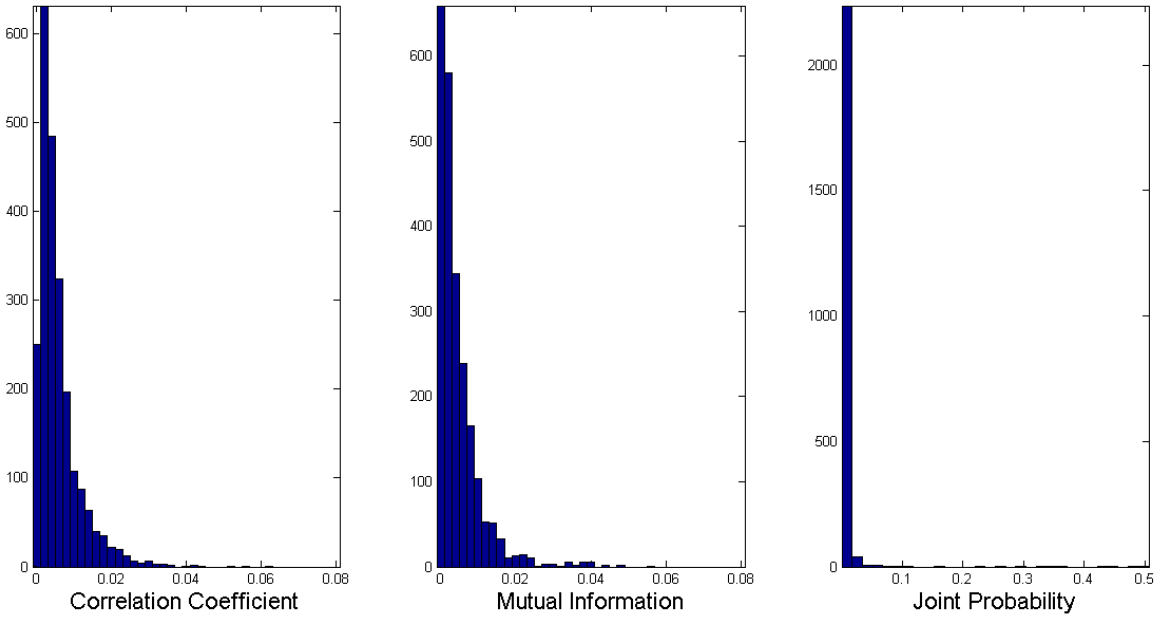

Figure 4.6: A histogram of the saliency score averaged over all trials. K =10

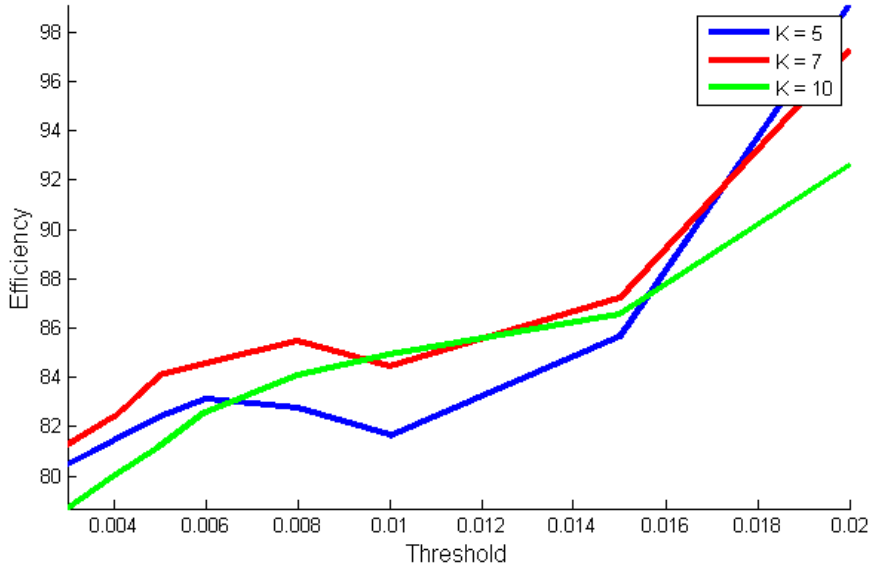

Figure 4.7: Performance of ANSWER for various K values. Weight type: Correlation coefficient

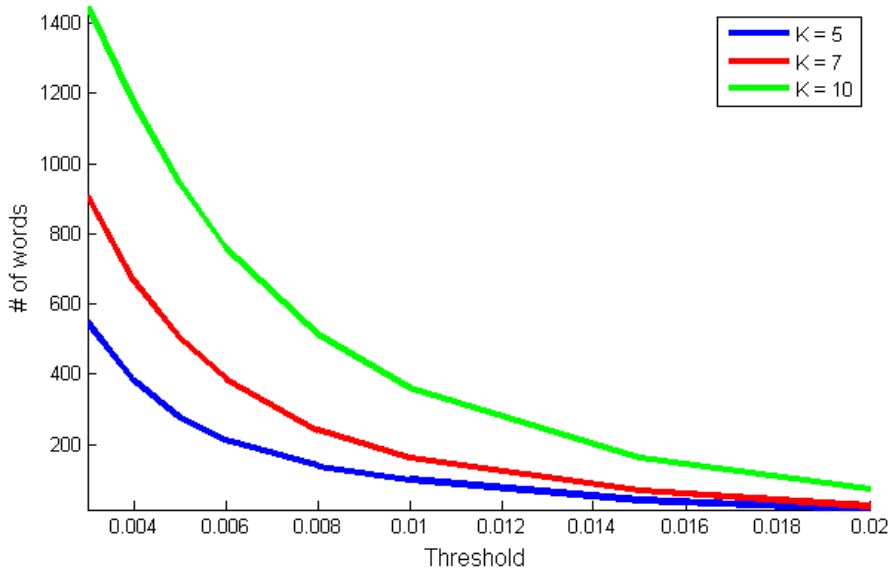

Figure 4.8: Length of word lists vs Threshold for different K values. Weight type: Correlation coefficient

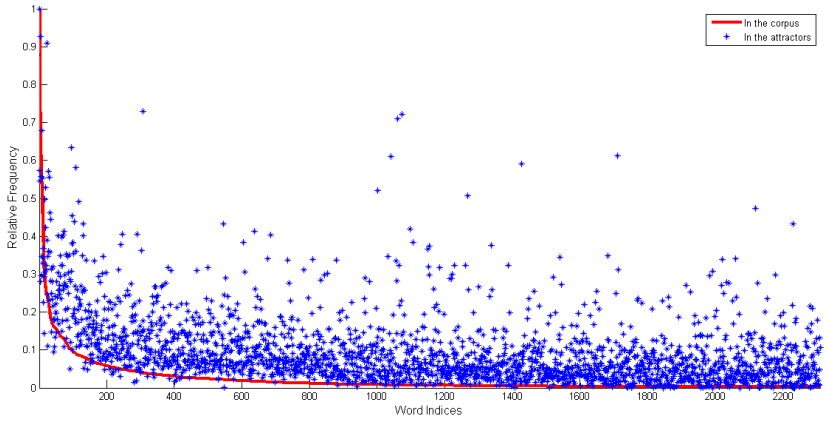

Figure 4.9: Relative corpus frequency and relative attractor frequency for words (averaged over 7 trials), Weight metric: Correlation coefficients K =10

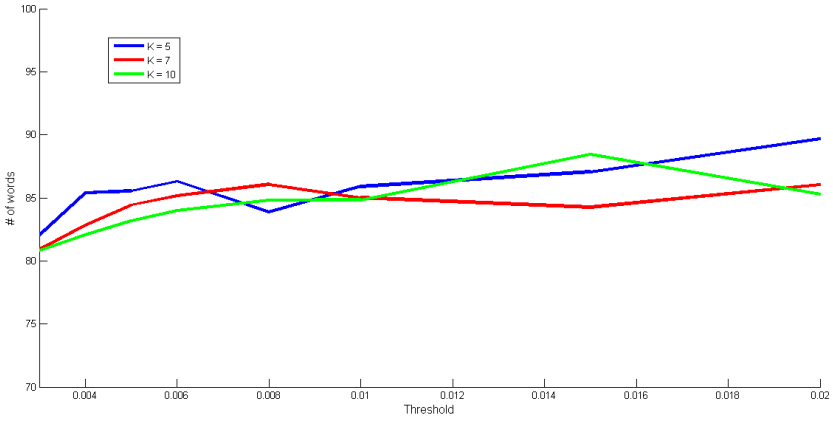

Figure 4.10: Performance of ANSWER for various K values. Weight type: Mutual Information

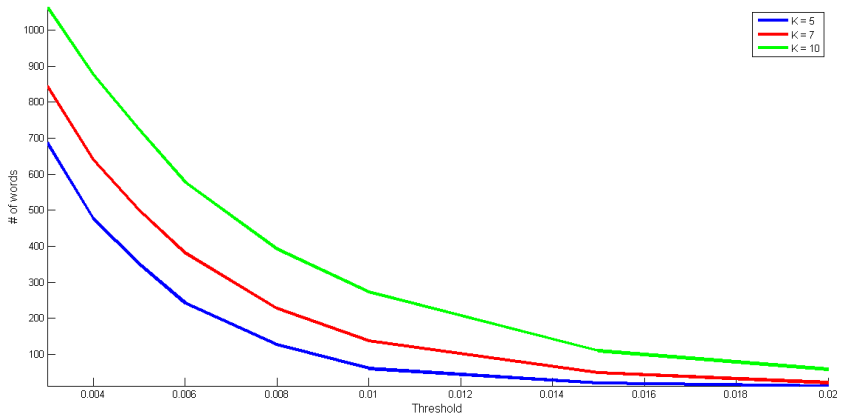

Figure 4.11: Length of word lists vs Threshold for different K values. Weight type: Mutual Information

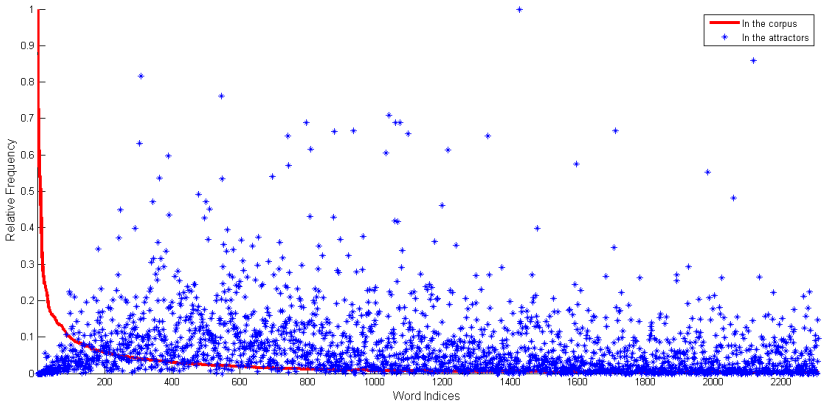

Figure 4.12: Relative corpus frequency and relative attractor frequency for words (averaged over 7 trials), Weight metric: Mutual Information K =10

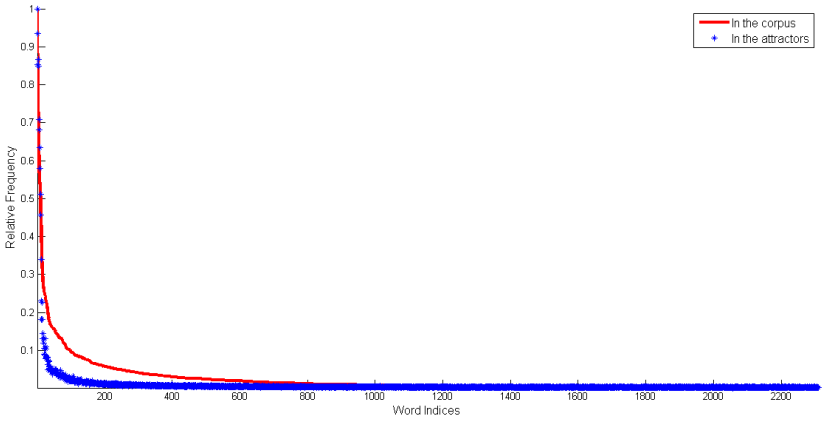

Figure 4.13: Relative corpus frequency and relative attractor frequency for words (averaged over 7 trials), Weight metric: Joint Probability K =10

4.1.2 Discussion

Cueing the network with 5000 random cues over 7 trials generated 35,000 ideas. The saliency score $S(v_i)$ for each word v_i was computed and words ranked accordingly. Based on a predetermined range of thresholds, words lists were drawn and those words deemed salient by ANSWER. Naturally, higher thresholds would yield shorter lists. The longer the lists, the greater the chance to include non-salient words. The length of the word list, N_trial, was used as the reference to draw word lists of the same length from the other metrics. Efficiency was calculated as the percentage of true positives that were picked up by each method.

Figures 4.1,4.3,4.2 show the distribution of the size of ideas for three representative combinations of K and weight type for the IJCNN corpus. As can be seen, most of the ideas are of the nominal desired size, K, but there is a small number of ideas that exceed this size because of the soft-competitive rule. There are no instances of ideas smaller than K because neural activity never falls below 0. Figures 4.4,4.5,4.6 show the distribution of the saliency scores of the words for all weight metrics and values of K – also for the IJCNN corpus. It

is noticeable that, while the correlation coefficient and mutual information weights lead to a fairly narrow saliency range, joint probability assigns very high saliency scores to some words. This reflects the fact that networks with joint probability weights tend to produce attractors biased heavily by word frequency, as will be demonstrated below.

The results for each weight type are discussed separately next.

Correlation Coefficients An example of the attractor sizes for this case have already been shown in Figure 4.1 and the first plot in Figures 4.4,4.5 and 4.6 show that the shape of the distribution of saliency scores remains similar implying independence of performance on K. The dependence of the network's performance on K can also be studied from figure 4.7. The curves for different values of K have approximately the same shape indicating that the network is not significantly dependent on K in terms of performance. However, it can be seen from Figure 4.8 that as K increases a greater number of words clear the threshold, which is expected because K basically controls the number of words that are allowed to be part of an attractor. The threshold also plays an important role in determining the length of the list. As the threshold increases, the number of words yielded by ANSWER gets smaller. Figure 4.9 shows the relative frequency of words in the corpus versus the relative frequency of the words in the attractors. It can be seen that the general profiles of the two plots follow a similar shape and hence it can be said that correlation coefficient weights would pick up salient words with a preference for word frequency i.e. words that are seen more commonly across the corpus and are salient are given preference over salient words that are not as frequent. However, it can also be seen that there are many words with low relative corpus frequency having high relative attractor frequency. This is a desired trait because we do want the network to be able to pick up salient words that may not occur too frequently in the corpus. Comparisons between word corpus frequency ANSWER have shown that ANSWER is consistently better as a saliency detector. From Tables 4.1,4.2 and 4.3 it can be seen that in the case of correlation coefficients as the weight metric, ANSWER is able to

almost sustain and in many cases increase the concentration of salient words as the word lists gets shorter, reaching near 100% efficiency for a threshold of 2%. This can be appreciated better by observing the shape of the curves in Figure 4.7. Of course, increasing threshold also decreases the number of salient words returned, so the list for high thresholds is best seen as a highly selective set of important words. With respect to ANSWER's performance against other techniques, from the presented data it can be seen that in almost all cases ANSWER's word lists are more densely populated with salient words irrespective of the length of the word list extracted. The results shown in these tables are averaged over 7 trials and the standard deviations are given in table 4.10. Hence correlation coefficient is a very good candidate for weights in setting up ANSWER.

Mutual Information - The ANSWER-IJCNN network set up using mutual information has shown interesting dynamics too. Figure 4.2 show the distribution of idea sizes. Figure 4.10 shows the performance of network for different values of K. The fact that the performance stays quite consistent across the values of K that are chosen mean that the system is not dependent on K. Interestingly, from the same figure it can also be seen that the performance stays within the 80-90% range for all values of K at all thresholds. In other words, the system's performance remains consistent irrespective of the parameter K and the length of the word list that is drawn out of it, hence displaying remarkable stability. This is further proven using the middle plot in Figures 4.4,4.5 and 4.6 which show that the shape of the distribution of saliency scores does not change with K. However, at a given threshold value, the length of the lists for different values of K increases with K as shown in 4.11. This behaviour is also expected because higher values of K creates bigger ideas including more words. Figure 4.12 shows that the relative frequency or words in attracts is quite different from the relative frequency of words in the corpus, implying that ANSWER set up using mutual information weights orders words purely by its notion of saliency not influenced by word corpus frequency. In fact, it can be said that high frequency words are suppressed

from being chosen as salient. Tables 4.4 through 4.6 presents the performance of ANSWER-IJCNN with the mutual information weight type in comparison with other standard metrics. It can be seen that mutual information performs better at extracting longer lists but the "Node Degree" metric seems to perform slightly better when it comes to extracting shorter lists. The results shown are averaged over 7 trials and the standard deviations are shown in table 4.11.

Joint Probability - With the weight between neural units set as the joint probability of the occurrence of the corresponding words, the ANSWER-IJCNN network does not perform well compared with the other weight types or methods. From the third plot in Figures 4.4,4.5 and 4.6 it can be seen that the vast majority of the words have low saliency scores, and there are a few words with extremely high scores. For K =5, out of the 2309 words approximately 2200 words have saliency score less than 0.005 and there are about 10 words with saliency score greater than 0.1. This is because of the fact that the same set of words win consistently in the attractor dynamics. From figure 4.13 it can be seen that the saliency scores produced by the joint probability network almost exactly follow the word corpus frequency plot. The the system is biased strongly in favor of high frequency words, with most initial conditions leading to attractors with these words. This inhibits the ability to sample the rest of the semantic space for other ideas and other attractors. From tables 4.7 through 4.8 it can be seen that the performance of the ANSWER-IJCNN network with joint probability weights is only about as good as "Frequency". The results shown in these tables are averaged across 7 trials and the standard deviations are shown in table 4.12.

Overall, ANSWER has proven to be a viable and efficient solution to identifying salient words in corpora such as the IJCNN corpus, comprising a collection short and distinct documents that are all pertaining to a general common topic - in this case, neural networks. It can also be concluded that, in general, using pointwise mutual information weights leads to the best performance except when very short lists of a few salient words – such as keywords

– are needed, where correlation coefficient weights and even non-neural metrics such as node degree do better. Networks with joint probability weights perform poorly and are not recommended.

4.2 Origin of Species

Darwin's *Origin of Species* represents a corpus that can be categorized as a single long, technical document. Thus, it provides no natural mechanism for applying metrics such as TF-IDF. Preprocessing left us with a total of N_W =49,923 word tokens out of which after dropping words that had 4 tokens or less, N_V =2,749 unique words were left. The document contained N_S =5,391 sentences in all.

4.2.1 Results

As explained earlier, due the unavailability of a ground-truth salient word list, the performance on this corpus is only analyzed anecdotally and from response to surveys. K was set at 7 and ANSWER was set up based on all three weight types. Using the same procedures that were used with the IJCNN corpus, salient word lists from the other methods were obtained. Surveys were taken from 5 individuals who were familiar with the book and their responses were combined to compare performance based on the "Minimum Rule", "Majority Rule" and the "Maximum Rule".

ANSWER-C	ANSWER-M	ANSWER-J	Degree	Frequency	Betweenness centrality	Eigenvector centrality	Weights
species	thick	species	natural	natural	forms	forms	forms
natural	cattle	natural	species	forms	natural	natural	natural
forms	wax	forms	case	case	pliocene	species	selection
generally	arctic	case	generally	generally	generally	generally	species
selection	bat	generally	organic	selection	selection	life	life
varieties	level	selection	animals	varieties	case	modification	period
organic	prey	varieties	great	organic	plants	modified	inhabitants
life	sediment	organic	number	animals	animals	existing	number
period	sheep	animals	life	plants	varieties	animals	animals
number	tail	plants	selection	facts	profoundly	selection	modification
characters	beak	facts	existing	life	organic	widely	structure
existing	subsidence	life	certain	period	number	inhabitants	cells
structure	plains	period	believe	distinct	facts	certain	generally
groups	seasons	distinct	period	believe	period	great	change
new	bones	believe	long	number	existing	long	varieties
individuals	germinated	number	large	great	great	change	great
modification	comb	great	plants	characters	life	number	long
inhabitants	wall	characters	distinct	existing	characters	probably	modified
descended	quickly	existing	varieties	structure	structure	case	formations
long	tumbler	structure	probably	groups	distinct	period	place
change	feeding	groups	structure	new	believe	large	characters
common	feathers	new	facts	produced	individuals	successive	widely
crossed	fossiliferous	produced	almost	instance	modification	consequently	existing
modified	pistil	instance	closely	degree	groups	structure	plants
islands	limbs	degree	inhabitants	large	acclimatization	produced	bees
related	characteristic	large	modification	individuals	genera	continued	accumulated
variation	degradation	individuals	widely	important	produced	almost	new
parent	resist	important	far	modification	degree	nearly	increase
successive	contingencies	modification	produced	closely	view	individuals	individuals
formations	indispensable	closely	change	inhabitants	far	place	certain
inherited	building	inhabitants	individuals	view	long	vary	groups
place	alike	view	degree	descended	islands	new	wax
intermediate	fantail	descended	nearly	long	common	degree	birds

Table 4.16: Sample word list from different algorithms; ANSWER-C, ANSWER-M and ANSWER-J correspond to ANSWER set up with correlation coefficient, pointwise mutual information and joint probabilities, respectively.

	Minimum Rule	Majority Rule	Maximum Rule
ANSWER-C	73.02	95.24	100.00
ANSWER-M	69.84	96.83	100.00
ANSWER-J	60.32	88.89	96.83
Degree	52.38	79.37	96.83
Frequency	58.73	87.30	100.00
Betweenness centrality	60.32	85.71	96.83
Eigenvector centrality	65.08	87.30	98.41
Weights	71.43	88.89	98.41

Table 4.17: Relative Performance (as % of true positives) of ANSWER (with different weight types) vs other methods based on responses from surveys; ANSWER-C, ANSWER-M and ANSWER-J are same as 4.2.1

4.2.2 Discussion

Word lists shown in table 4.2.1 provide anecdotal comparison between salient words identified by different weight types with ANSWER and by other methods. From perusing the word lists, it can be seen that the relative performances in this corpus are similar to those for the IJCNN corpus. Word lists from correlation coefficient and mutual information seem to be made up of better words than other methods. Also, based on quantifying the responses to the anonymized word list surveys shown in table 4.2.1, it can be said that correlation coefficient and mutual information are good candidates for this application. By comparing those two word lists with the frequency based list it can be seen, as in the case of IJCNN, that correlation coefficient's bias towards high frequency words is also reflected in this corpus.

Origin of Species as a corpus is similar to IJCNN in the sense that it is also a technical document and different from IJCNN in that it does not contain distinct documents.

ANSWER has thus been demonstrated to be a successful tool in applying to the saliency detection task on such a corpus as well.

4.3 Playing It My Way

This autobiography of the great Indian cricket star, Sachin Tendulkar, was processed as a single document, and all three weight types were extracted. Pre-processing N_S =12,868 sentences, gave a total of N_W =56,302 word tokens from which N_V =1,897 unique words remained after eliminating words that occurred 4 times or less. The same random cueing-based simulation procedures as with the other corpora were carried out on this corpus.

4.3.1 Results

Just like the '*Origin of Species*, the performance of ANSWER and other methods on this corpus has been analyzed by looking at sample word lists and based on response to surveys. Performance metrics based on the "Minimum Rule","Majority Rule" and "Maximum Rule" have been computed using surveys taken from 5 individuals who were familiar with the book.

ANSWER-C	ANSWER-M	ANSWER-J	Degree	Frequency	Betweenness centrality	Eigenvector centrality	Weights
played	dressing	played	time	india	ball	time	played
india	ground	india	bat	match	time	ball	bowling
match	pitch	match	match	test	bat	bat	match
test	stump	test	day	time	bowling	bowling	bat
ball	injury	time	ball	ball	match	just	cricket
runs	fans	ball	started	bat	started	come	series
series	outside	bat	bowling	runs	day	team	team
bowling	umpire	runs	cricket	cricket	good	day	time
wickets	leg	cricket	team	day	asked	started	day
won	normal	day	final	started	cricket	good	wickets
bowlers	swing	started	just	series	team	game	players
south	incident	series	end	bowling	wickets	players	bowlers
world	caught	bowling	good	team	players	asked	asked
room	served	team	game	wickets	australia	final	room
hit	vinod	wickets	did	scored	just	home	good
africa	caused	scored	runs	final	end	make	hit
cup	body	final	come	just	game	end	stump
west	involved	just	series	game	final	way	world
sri	andrew	game	asked	sr	bowlers	match	test
fast	bcci	sr	second	end	dec	came	started
dressing	drink	end	way	tendulkar	england	second	game
lanka	midwicket	tendulkar	make	did	make	bowlers	just
indies	square	did	players	australia	got	cricket	come
stump	sunil	australia	wickets	good	came	room	runs
st	symonds	good	managed	won	second	field	end
drawn	tears	won	got	second	runs	helped	fast
nd	national	second	went	way	managed	got	got
outside	slightly	way	scored	went	scored	series	helped
leg	waving	went	bowlers	asked	come	managed	home

Table 4.18: Sample word list from different algorithms; ANSWER-C, ANSWER-M and ANSWER-J correspond to ANSWER set up with correlation coefficient, pointwise mutual information and joint probabilities respectively.

	Minimum Rule	Majority Rule	Maximum Rule
ANSWER- corr	74.29	94.29	97.14
ANSWER - mutualInfo	68.57	91.43	100.00
ANSWER - jointProbs	62.86	80.00	85.71
Degree	60.00	74.29	88.57
Frequency	62.86	80.00	94.29
Betweenness centrality	54.29	74.29	85.71
Eigenvector centrality	51.43	71.43	85.71
Weights	68.57	82.86	94.29

Table 4.19: Relative Performance (as % of true positives) of ANSWER (with different weight types) vs other methods based on responses from surveys; ANSWER-C, ANSWER-M and ANSWER-J are same as 4.3.1

4.3.2 Discussion

Again, ANSWER has performed well with correlation coefficient and mutual information weights. Anecdotal results have been shown in 4.3.1 and efficiency as a percentage of true positives computed from survey responses is shown in table 4.3.1.The survey was taken on lists with the top 36 words from each algorithm. Consistent with performance on other corpora, correlation coefficient has a frequency bias.

This corpus, like *Origin of Species*, is a single document, but unlike both IJCNN and *Origin of Species*, it is a non-technical corpus. Also, it is a contemporary work that includes a lot of slang and casual vocabulary. ANSWER has worked well on this corpus and can be a reliable tool to identify salient words.

4.4 Comparison of Weight Types

While it is apparent that joint probability is not a suitable type for salient word extraction, correlation coefficient and pointwise mutual information provide some interesting results. Tables 4.13, 4.14 and 4.15 show the performance comparison between the three weight metrics in the IJCNN corpus. Figures 4.4,4.5 and 4.6 show that while all these methods are quite robust to changes in K, only correlation coefficient and mutual information provide a good distribution of saliency scores so as to draw lists of various lengths at different thresholds. Joint probability assigns very high scores to high frequency words which makes it an undesirable choice of weight metric. The choice between correlation coefficient and mutual information weights can be made based on the following two key issues:

1. *Frequency Bias*: As discussed in the results of the IJCNN corpus, from figures 4.9 and 4.12 it can be seen that correlation coefficient weights prefer words that occur with high frequency in the corpus. Pointwise mutual information, on the other hand, suppresses high frequency words and prefers medium frequency ones. This quality of mutual information might come in handy if we do not remove stop words from the corpus. Articles and prepositions that are extremely high frequency would probably be ignored by this method. This can also be observed from the word lists of the other corpora, results for mutual information do contain salient words but those are not the highly frequent words that the authors use.

2. *Concentration of Salience*: As the threshold increases and as smaller and smaller word lists are extracted, correlation coefficient increases the density of salient words in the list. It acts as a gradual concentrator of salience whereas pointwise mutual information is consistent in its performance. From tables 4.13 through 4.15 it can be said that pointwise mutual information is a good metric for extracting longer lists, whereas correlation coefficients would work better for shorter lists.

ANSWER was developed and tested with three different types of corpora – a collection of technical abstracts, a technical book and an autobiography – and it has proven to be an effective method for the word saliency detection task. Three different weight types were tried and their pros and cons have been discussed. Designed as a neuro-cognitive model with an intention to develop a generic model for semantic cognition, ANSWER has performed quite well in comparison with other industry standard saliency detection methods.

Chapter 5

Conclusion and Future Work

We have applied an unsupervised attractor network-based approach for detection of salient words in text corpora to - (1) a labelled corpus derived from abstract for IJCNN 2009, 2011 and 2013; (2) *Origin of Species*, a technical document by Charles Darwin in the form a book, and (3) A recently published autobiography of Sachin Tendulkar, *Playing It My Way*. The results reported here show that ANSWER did at least at well and usually better than the other simple saliency detection metrics that were tried. The proposed algorithm has the advantage of being applicable to undifferentiated corpora, and of requiring specification of relatively few parameters. Different weight types have been explored and recommendations for choice of weight type have been made.

The process of identifying salient words using the network is a two step process - first, the text corpus is pre-processed to remove well known stop words and retrieve statistics from word occurrences and second, an attractor network is used to filter the words further and come up with a list of salient words. To summarize, the basic features of ANSWER are as follows:

- ANSWER models a neurodynamical approach to semantic networks and attractors represent ideas that are emergent from the competitive actvityt in the recurrent asso-

ciative network.

- The network can be modeled by a few parameters – K, α and γ. Of these, α and γ need not be altered. K is the primary parameter and it's effect on the results has been shown.

- ANSWER provides a completely unsupervised approach to saliency detection and is hence a more practical tool for this application.

- The ability of ANSWER to identify salient words appears to be independent of the nature and structure of the corpus.

Overall, while the results with ANSWER were encouraging, there are some challenges that remain and there is a lot of scope for further improvement and expansion. Some directions along which ANSWER can be further explored are:

- Exploring other ways to specify association weights in the ASN, e.g., using other weight types or linear/non-linear combinations of the currently used metrics.

- Eliminating the first stage of removing known non-salient words. If the stop word removal step is not required, ANSWER could be applicable in domains where comprehensive stop word lists are not available and perhaps even to texts in other languages.

- Modeling ANSWER in a hierarchical network of networks to combine attractors from different weight metrics at a higher level of abstraction.

- Combining multiple ANSWER and non-ANSWER heuristics in a mixture-of-experts (MOE) setting.

- Applying ANSWER and MOE approaches to related problems such as named entity extraction, topic extraction, document classification, keyword identification, search query generation, text summarization, etc.

Bibliography