The incorporation of human memory models into artificial intelligence-based problem solving

by

Justin W. Deschenes

Bachelor of Computer Science, University of New Brunswick, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Michael Fleming, Ph.D., Computer Science
Examining Board:	Patricia Evans, Ph.D., Computer Science, Chair
	Kenneth Kent, Ph.D., Computer Science
	Biljana Stevanovski, Ph.D., Psychology
External Examiner:	

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

September, 2009

Dedication

To Danielle, Keifer and all who have helped along the way.

Abstract

Artificial intelligence is in an ideal position to use research in other fields to improve itself. In this thesis, ideas from psychological and cognitive models of learning, as well as human memory models, have been hybridized to form a problem solving system. This was done by applying aspects of human learning and memory to a problem in natural language processing, the word sense disambiguation problem. The question being answered is "can human-like memory improve accuracy in a particular area of artificial intelligence, specifically word sense disambiguation?". This involved exploring the various models and presenting how portions of the models could fit together into a computer system. The model was implemented and the details for how it came together, what worked and what did not are presented. Finally, the system was tested, using an open source competition data set, and compared with other systems using the data set. The results ranked the system among the top five contestants. The results also show a potential for future accuracy improvements in determining word sense, as well as a chance to better model human memory and learning.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The pursuit of better artificial intelligence has brought an analytical eye to numerous fields and disciplines. While the idea of computers or automata emulating human intelligence is as old as the greek myth of Hephaestus and his bronze man Talos[13], the difficulty of the problem has prevented the realization of the true "Artificial Man". Research into math, statistics, logic, and general computer science has improved the overall effectiveness of the field. Matching various techniques to sub-fields within artificial intelligence has further improved specific areas and cases (e.g. formal grammars and Natural Language Processing, symbolic logic and constraint problems, statistics and inference techniques). While current improvements have been related to better statistical algorithms, more descriptive mathematical models and more powerful computer hardware, the necessity to try other techniques or reinvent old techniques has emerged as the field matures. Although human

sentience is largely a mystery, our basic problem-solving abilities offer a great model for an intelligent framework.

When searching for a problem domain for this thesis, it was necessary to find a problem that we (humans) are exceedingly well suited for. While there are numerous areas where a human outmatches a computer, the area had to have a well defined problem and be easily testable against non-human artificial intelligence systems. We are supreme fuzzy pattern matchers. Even today, no computer is as capable at matching a name to a face as quickly as we are. Pattern matching relates to our language, humour, recent events, metaphors, expressions, context, and double meanings. The human brain handles and works with ambiguous situations and statements extremely well. With this last point in mind, natural language processing became a good candidate and the specific task of word sense disambiguation seemed to have a well defined problem space. As the title of the problem suggests, word sense disambiguation relates to the action of understanding which meaning an ambiguous word has in the context of a particular sentence. Current artificial intelligence systems perform well below the required effectiveness to make adding this to existing technologies useful.

The motivation for this thesis is two-fold. The main focus is to adapt human intelligence and the human learning process, specifically human memory models, to an artificial intelligence system. This would involve looking at various models and taking what is useful or fits well with other systems and combining them to promote successful problem solving. The second focus

is to show that this adapted system can be useful in solving hard artificial intelligence problems, specifically word sense disambiguation. Recently, discussion in some of the academic journals put forth the idea of using the latest ideas in biology, social science, psychology, and cognitive science to improve aspects of artificial intelligence. Looking into the background of the word sense disambiguation problem, there was no indication of work that used even a basic biological model. With this gap, and the prospective good match of the human intelligence framework or the human memory model to the word sense disambiguation problem, it seemed like a prospective avenue to further the field. Work done in this area can open up a few interesting avenues. First, if it proves interesting or useful in the general case, it promotes another method for improving current artificial intelligence systems. Second, if it proves useful specific to the word sense disambiguation problem it could help improve the field and help other areas within natural language processing. Not only that, even with a partial success it would be useful to identify where in the field of artificial intelligence the human models could make an impact, and to apply it. Finally, if little use comes out of it, or the implementation of the human memory model is flawed, a researcher may gain inspiration from the ideas presented.

Artificial intelligence research has shown that statistical methods perform well in general; however, as the data sets grow and as the problem becomes more general, often correct answers and small patterns get lost in the noise. A number of techniques have been developed to combat this problem;

when these techniques are combined, small incremental gains in accuracy are achieved [33]. Modelling human memory may provide insight into improving accuracy when working with both large data sets and loosely coupled data areas. Humans using general intelligence, past memories and other conscious and non-conscious mental abilities are able to instinctually discover minute patterns in large sets of data, quickly [21]. Computer algorithms attempt to statistically determine patterns; however, the ability to repress certain patterns, while weighing other patterns more heavily, is more effectively done by a human. This ability allows for humans to cut out a vast majority of the data set through the use of logic, experiences and understanding. These qualities certainly require more than just memory; however, without memories, none of these abilities could occur.

It is hypothesized that the incorporation of human-like memory should provide improvements in the area of word-sense disambiguation. The problem requires general knowledge and the ability to recall information with a bias dependent on the context of each paragraph, sentence and word. Developing human-like memory in concert with current statistical techniques should provide more accuracy in word sensing and provide insight into the impact of imperfect memory on difficult, general problems.

The question then is: can human-like memory improve accuracy in a specific area of artificial intelligence, specifically word sense disambiguation?

Chapter 2 will present some background information on artificial intelligence, natural language processing (specifically the word sense disambiguation prob-

lem), and early artificial intelligence work done in biological systems. It will also give an overview of some relatively modern research in human memory models that will be adapted and used throughout the thesis. Chapter 3 discusses the methodology of the theoretical models chosen. This chapter illustrates the "why" of the models, it highlights what could be potentially useful in its implementation and gives insight into design considerations. Also presented are what areas are lacking or missing and what has been done to fulfill the requirement, while outlining what is out of scope or too difficult to implement. Chapter 4 presents the implementation details, or the "how". This chapter shows the details of the test implementation, the tools used, the data set, and the algorithms used to implement the chosen theoretical models. Chapter 5 presents and discusses the quantitative and qualitative results. Finally, Chapter 6, the conclusion, discusses the overall success of the work presented and how it relates to the two-fold motivation and goals mentioned earlier in this chapter. It also puts forth interesting avenues for future areas of research on the topic.

Chapter 2

Background

2.1 Natural Language Processing

Theorists have posited that true intelligence would allow computers to carry on insightful conversations. The conversations would be of such high quality that the human would be unable to determine that they were in fact communicating with a computer[39]. Natural language processing(NLP) is theorized to be AI-Complete [24], meaning that to solve this problem would give computer scientists the insight required to create computer programs as intelligent as humans. Initial work in this area showed remarkable success: in automatic translations between languages [23], the application to restricted words and languages[36] all provided good results. However, application to the general problem of linguistics was much slower. Toy programs were developed that simulated conversation; these chatterbots could be programmed

to resemble intelligent beings. However, the programs lacked the knowledge depth and application required to be considered truly intelligent, instead relying on canned responses.

NLP requires numerous approaches and techniques to evaluate human texts. Humans often require multiple types and levels of processing to correctly comprehend language [27]. Most of these levels are used together to differentiate between the various meanings, as each of the levels adds a method of disambiguation to the context of the linguistic structure. NLP programs often use multiple levels and combinations of linguistic analysis; it is in this that most programs differ [27]. The various approaches to analyzing NLP include the following.

- The symbolic approach: This often involves the use of a symbolic rule system developed by humans with an emphasis on known representation and language processing algorithms.

- The statistical approach: This uses observable information modelled from the linguistic world based on probabilities, commonly using Hidden Markov Models.

- The connectionist approach: This combines statistical approaches with various components of representation, allowing manipulation of the underlying models and sub-problems within NLP, while continually improving confidence in answers.

2.1.1 Word Sense Disambiguation

One of the sub-problems of natural language processing is word sense disambiguation (WSD). WSD is the act of identifying the meaning of a word in a sentence when the word has multiple meanings. An example is in the following two sentences: "I heard a loud pop.", "I enjoy an occasional pop.". In the first sentence, the word "pop" refers to a noise; in the second sentence, "pop" refers to a drink. Humans, with the use of a large volume of collected information and the skill set developed by working in the ambiguities of language, are remarkably effective at sorting out the meaning of the words. Because of the general intelligence required to solve these problems, this problem is also classified as AI-Complete [8]. The problem was encountered initially in automated machine translation; in fact, it was one of the first "hard" road blocks [31]. Initial approaches considered using computers to automate the understanding of languages; however, at the time, there was an insufficient collection of data [31]. In the 1990's with the prevalence of the internet came a large collection of electronic documents. With computational power also increasing greatly over the previous twenty years, statistical and computationally expensive techniques were implemented and allowed for improved accuracy. From the Senseval/Semeval contests [1], we are able to see how various researchers performed. The Semeval (earlier competitions were called Senseval) is a contests which has outlined a number of challenges over the years; these challenges included different dictionaries and different goals as related to word sense disambiguation, and semantic word problems. The

trends from the competitions are quite difficult to determine; however, on the whole, performance has slowly increased from contest to contest. Successful techniques in WSD include memory-based learning [16, 22, 40], instance based learning [30], decision lists [41], ensemble methods [18], kernelization [20], as well as various knowledge hybrid methods [34].

The last forty years in artificial intelligence have seen the rise of statistical techniques, which proved to be useful tools for a number of hard problems. Over these years, both technology and theories in related fields have come a long way. It is these new theories and tools that will be used to challenge ideas and techniques that have held true in artificial intelligence for years. Biological theories about the inner workings of the brain show promise for a number of areas of computer science, from parallel architectures to general intelligence [35, 2]. The investigation into biologically inspired cognitive architectures opens up a number of areas [37]. Just as early computer scientists looked to the brain in creating computational and artificial intelligence models, again are a number of AI researchers looking towards the theoretical aspects of the brain and mind. New theories into cognitive awareness, thought processes, learning, memory, understanding, self-examination, along with improvements in neuroscience, brain chemistry understanding and psychology, allow for a number of possible research avenues.

One area that has a rather small volume of work in computer science, but has a remarkable importance in biology, psychology, neuroscience and learning science, is that of human memory. Three interesting human memory models

come from the psychology and cognitive science fields. The theories are briefly described in the following sections.

2.2 Human Memory Models

2.2.1 Baddeley and Hitch

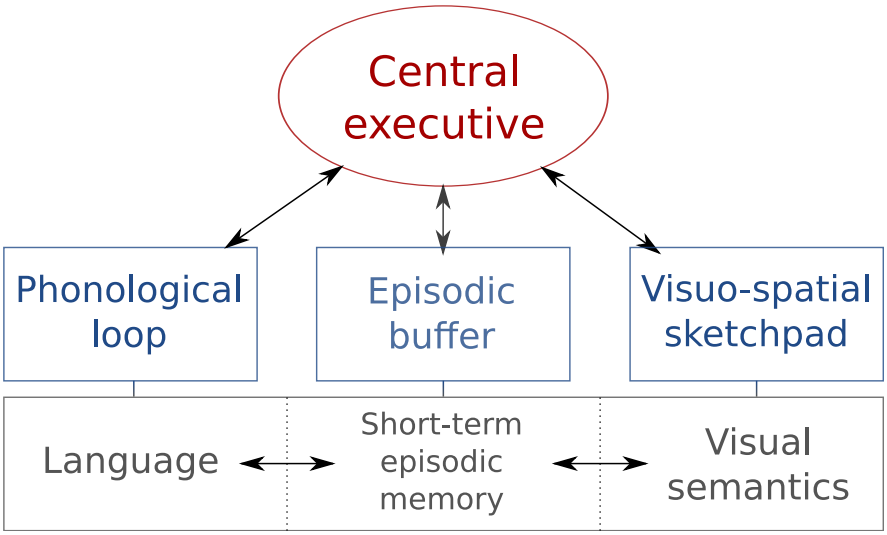

Figure 2.1: Baddeley and Hitch Model

In Baddeley and Hitch's model [6, 7], three systems (the phonological loop, the visuo-spatial sketchpad and the episodic buffer) are responsible for short-term maintenance of their respective information, whereas the fourth system, the central executive, is responsible for information integration and system coordination. The central executive system is responsible for directing attention to relevant information, suppressing irrelevant information and unnec-

essary actions, and providing the ability to multitask. Furthermore, the central executive also provides a supervisory system, which can guide the other components of working memory (part of short-term memory which handles immediate conscious processing) back into a stable working environment. The phonological loop takes auditory verbal information and written language and encodes them into an auditory code. This information is then repeatedly cycled in temporal order to refresh contextual information and prevent decay. The visuo-spatial sketchpad consists of the "visual cache" and the "inner scribe". The visual cache stores information relating to form and colour, where the inner scribe stores information relating to movement, speed and distances. orking memory allows manipulation of this environment to enable planning and spatial understanding. The episodic buffer provides a temporary memory component, which integrates the loop systems and other acquired information, producing a "unitary episodic representation"[7]. Through the chronological sequencing the applications of both loops are more easily transported to long-term memory. Furthermore, in uniting the visual and sound loops with chronological ordering, previously unknowable semantic information may present itself.

2.2.2 Cowan

In Cowan's theory [14, 15], working memory is part of long-term memory. The representation used by working memory is a subset of memory elements stored in long-term memory. Working memory has two levels. The first

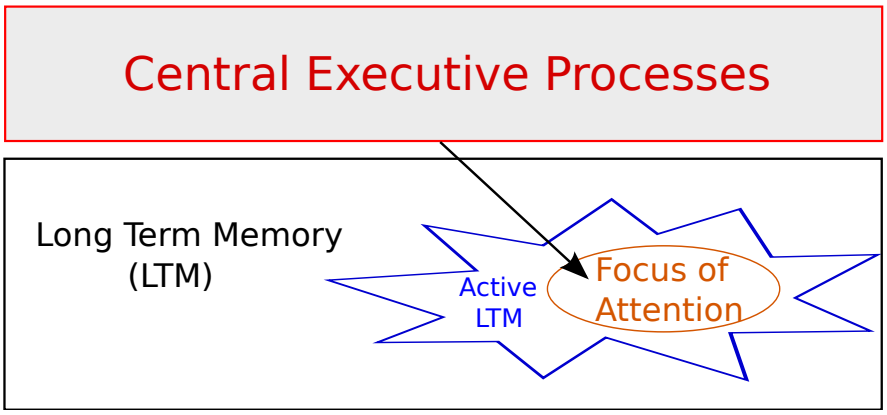

Figure 2.2: Cowan Model

consists of activated long-term memory elements of which there can be an endless number, as long-term memory is limitless. The second level is the focus of attention. This focus of attention has a limited capacity and holds a more detailed component on which work can be done; this level is able to contain about 4 "chunks" of information. Other cognitive scientists have added a new level to Cowan's original theory, which has increased focus and is only able to contain a single element [32]. The focus aspect of Cowan's theory has a few properties. These include:

- Focus is controlled by both voluntary (conscious) and involuntary (unconscious) methods.
- Focus is required to perform many day to day tasks such as: planning, understanding, and cause and effect.
- The level of control of focus and capacity of focus differ on an individual

basis.

This model readily lends itself to solving problems in both a high-level idea-oriented situation and a low-level detail-oriented situation, with the ability to change between the two. Working on and modifying long-term memory "chunks" in working memory allows for a useful instance-based memory component that can be used to focus on solving problems and understanding. Each application of problem solving then modifies the long-term memory instance it had used.

2.2.3 Ericsson and Kintsch

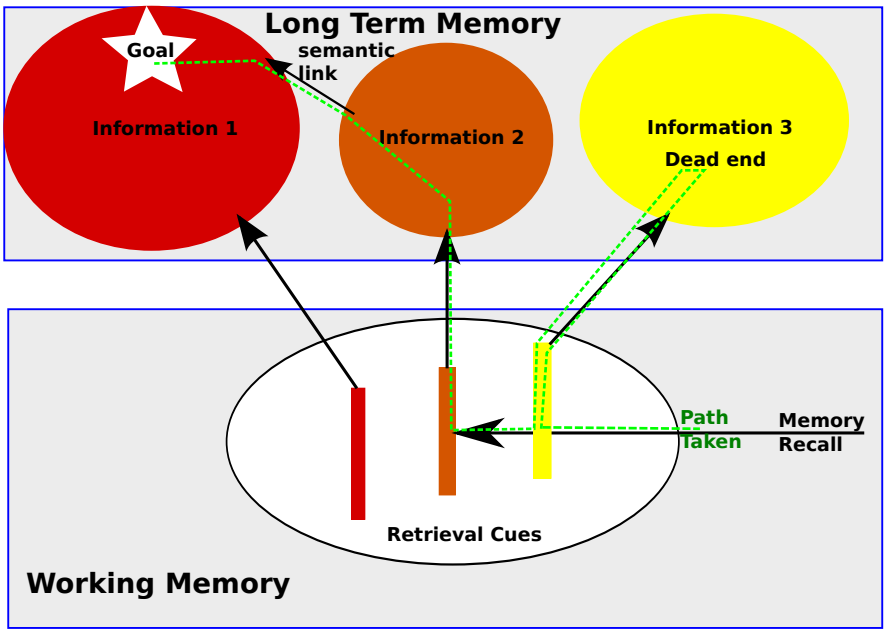

Figure 2.3: Ericsson and Kintsch Model

In this theory, Ericsson and Kintsch [17] hypothesized that humans use specialized memory for most tasks. They also suggest we store most of what we read in long-term memory, linking it through retrieval structures. Concepts can be held in working memory and can act as cues that link to the retrieval structures and return the memories associated. Different types of retrieval structures include generic, domain knowledge, and episodic text. Generic retrieval occurs deliberately, domain knowledge is retrieved through pattern matching and schemas which allow organization, and episodic text is retrieved through text comprehension. Specialized memory is based on using specific knowledge of the area of memory and applying it to the storage and encoding of information in long-term memory. The cues act essentially as a "hash", or quick lookup of the long-term memory connected to it; this allows great memory performance by allowing rapid and flexible long-term memory access [17]. Although the exact mechanism for cue creation is unknown, the retrieval cue must be encoded to handle temporal information. Newer information must have more of a precedent than older information, allowing memories and their structure to change as new information is presented. Encoded information is not only linked by a retrieval cue, but also has semantic links connecting the larger pieces of information to one another.

2.2.4 Additional Models

In addition to the three well known theories mentioned above, one area of cognitive science that may benefit memory research, but is notoriously difficult

to model and understand, is that of the unconscious mind. The importance of the unconscious mind can be summed up by a question asked by Greenwald, Klinger and Schuh [21]: "How good is the mind at extracting meaning from stimuli of which one is not consciously aware?" with answers ranging from: "it is simplistic and dumb" [28], to, "it is prevalent and strongly influences nearly all higher mental processes" [10]. The effect of the unconscious has in its majority created the complex and intelligent design necessary for living things through adaption and natural selection [9]. While the extent to which we rely on the unconscious region of our mind is debatable, the desire to understand how this part of our mind works, and its limits, can be found in everything from business management journals to the New York Times best-seller list. In an experiment, Bechara et al.[12] were able to show that participants in a rigged game were able to decide upon the advantageous strategy well before they could consciously understand why the strategy was successful. This unconscious understanding, or implicit memory, is a relatively new area of research, with almost all studies completed within the last twenty-five years. Current research suggests that this memory relies on an implicit, unconscious piece of knowledge previously learned and that the processes and structures are not completely related to explicit memory [38]. There has been some work in applying cognitive models to aspects of computer science and artificial intelligence. The majority of these systems focus on a symbolic rule-based interpretation of memory and are designed more from a cognitive psychology point of view. Two popular systems are the

ACT-R and Soar architectures.

ACT-R (Adaptive Control of Thought - Rational) began in the early 1970's where it added a rule-based or procedural system to the basic declarative models of the time [3]. The work done over the last forty years added a mathematical rational framework [4], created specialized models to simulate brain functions [5], and added constructs to reflect cortical activity [5]. ACT-R's approach to memory is to have two systems: declarative and procedural memory, where the procedural memory essentially acts like a neural net subsystem.

The Soar (State, Operator and Result) architecture, a symbolic cognitive architecture developed from the late 1980's with a goal of cognition and general intelligence [25]. While it does not incorporate all the models listed above, it does use working memory in conjunction with various forms of long-term memory and problem space searching in its use as a production rule-based system[26].

The Baddeley and Hitch theory takes the importance of bio-feedback loops and applies them to long term and working memory, adding a unit that controls overall interaction. This theory briefly mentions the importance of attention; however, the Cowan model expands on what attention and focus are and their importance to our memory. The Cowan model also attempts to explain how our conscious mind and memory interact, whereby greater focus on an element provides more information to the preclusion of other memories. The Ericsson and Kintsch model adapts the memory component

itself and its association with other memories. This allows for efficient storage and retrieval of memories, as the memories can be saved in different ways according to the way they were retrieved and extracted. Ericsson and Kintsch also help to explain how the mind and memories perform some tasks so quickly, with the retrieval structures between working memory and long-term memory and semantic links between memory components in long-term memory. Finally, the unconscious mind handles non-directed learning and its interaction with memory, decision making and understanding. While there is much debate in how the unconscious mind works, decision making and problem solving do improve when unconscious directions are considered.

Chapter 3

Methodology

This chapter discusses biological and psychological models and constructs that may have a positive impact on natural language processing and the word sense disambiguation problem. This chapter outlines the current system, the design decisions required by the models, and the architectural decisions required to make the systems work together. An attempt is made to show the number of design decisions that have to be made to balance the two accuracy goals of this thesis, the accuracy of the results and the accuracy of the model. Further details include what changes were required to get the models to work together, how the current system incorporated all the components together, and what is lacking in the current system. The implementation details of the various algorithms, structures and components are left to Chapter 4.

In the general case, psychological and biological models related to the human brain and its learning capability lay out various "executive" functions. These

functions are high-level objectives that, when intelligently combined, allow the mind to recall past events, solve problems, and learn. The executive functions themselves are made up of many other "building block" functions that support the goal of the executive functions and allow generic and adaptive problem solving.

Much about the mind's systems is a mystery to researchers, with insights coming from logically deductive models. Models are reviewed by the scientific community, and those that are perceived to work with what is known are given more confidence. With little describing the building block functions, coming up with a solution to all of the minor problems in this thesis's system was based on simplicity, good software practices and artificial intelligence research.

Through the course of the thesis work, there was a continual readjustment of the tradeoffs. The competing design points were: complexity, implementation time, faithfulness to the models and, when results became available, accuracy in the results. The initial designs were too complex and required too much time for little gain in either of the accuracies. After a few iterations of refining the core idea and removing some of the parallelism complexity, a plausible model was designed and implemented.

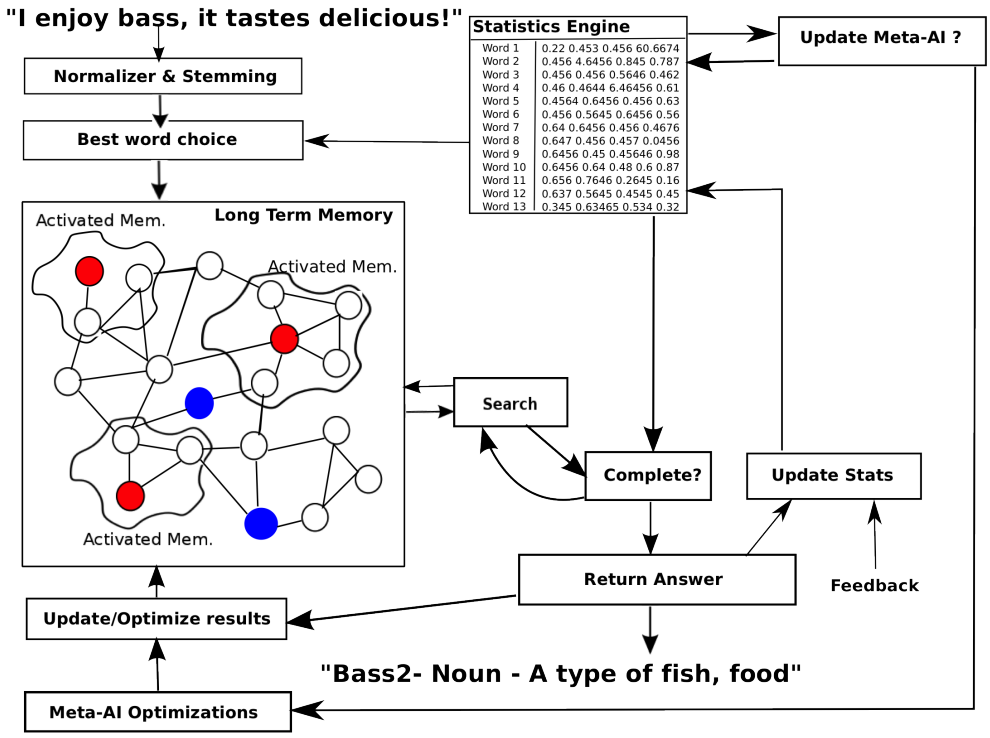

Figure 3.1: Proposed Design

3.1 High-level solution

Figure 3.1 gives a high level overview of the learning process. The process starts by passing in a sentence and a target word within the sentence that requires disambiguation. The sentence is then pre-processed in an effort to simplify data storage and to lower algorithm run-times. The process involves removing stopwords, then normalizing and stemming the remaining words. Stopwords are words that can be removed from processing as they add very little to the sentence. Normalizing is the removal of punctuation and case differences. Stemming is the act of generalizing words to their root form. In the sentence given in Figure 3.1, "I enjoy bass, it tastes delicious!", an example of stopwords are "it" and "I". The remaining words would be transformed to lowercase, punctuation would be removed, and the sentence would be passed to the stemmer. The output would be: "enjoy bass tast delici". This sentence is more compact than the original and still captures the essence of the sentence. The new sentence is then run through a "best word choice" algorithm. The "best word choice" algorithm uses previously collected statistical information to select the words in the sentence that have the greatest chance of successfully completing the disambiguation. This is done by determining the frequency with which each word in the sentence has appeared with the target, multiplied by the number of correct disambiguations the word has been a part of. After the best words are selected, they are loaded into a memory data structure. This data structure acts as working memory, or a

solution scratchpad for the problem instance. The memory paths used to solve the problem are recorded, with the goal of adding useful information to the long-term memory structure, to better future queries. The working memory connects the current problem to existing problems. The underlying structure is a graph, where the nodes represent words, and the edges are weighted to promote the best path to search the graph for a word sense. The working-memory graph consists of the current words, and adds the path each word-node used to attempt to disambiguate the target word. This information, along with information from the statistics engine, and the outcome are used to incorporate the working memory graph into the long-term memory graph. The memory search method for disambiguating the target word is paired with a fast statistical algorithm and make up the two arrows in Figure 3.1 pointing to "Complete?". The fast statistical algorithm uses a simple frequency analysis of target word to word sense, essentially returning the most commonly seen word sense for a given word. This method for determing word sense acts as the quick gut check. When these two functions have completed, they return a word sense and an associated confidence score. These two sets of results are then passed to the "return answer" section which uses the meta-component to determine which word sense is to be the final answer and what aspects of the contents of working memory, combined with the feedback from the answer given, should be used to update the statistical and long-term memory component. After the process is complete, a check is done in the statistical engine to determine how well the system is optimized,

seen in the "update Meta-AI?" box. If the check determines that changes should be made, the system is modified and testing resumes.

3.2 Current Model

The high level solution mentioned above gives an executive view of the system. This section goes into detail of what the components are, what design decisions were made, and how the system differs from the ideal solution.
The current model consists of three functional components and a higher level container object. The Statistical, Connective, and Meta-learning components are used by the Memory component to provide access to the various "executive" building blocks.

3.2.1 Memory object

Seen in Figure 3.2 the memory object could be considered the interface to learning. It is composed of the "Learning Environment Data Structure"(LEDS), which is the per-problem scratch pad or working memory, and the interface that communicates with activated long-term memory and the statistical component. Working with each of the components, it captures how the problem was solved and what was used to solve it. The information in the LEDS is then used by the meta-learning component to transfer knowledge gained in the instance to long-term memory.

1. a way to load and save the state of the mind. This way, the learning

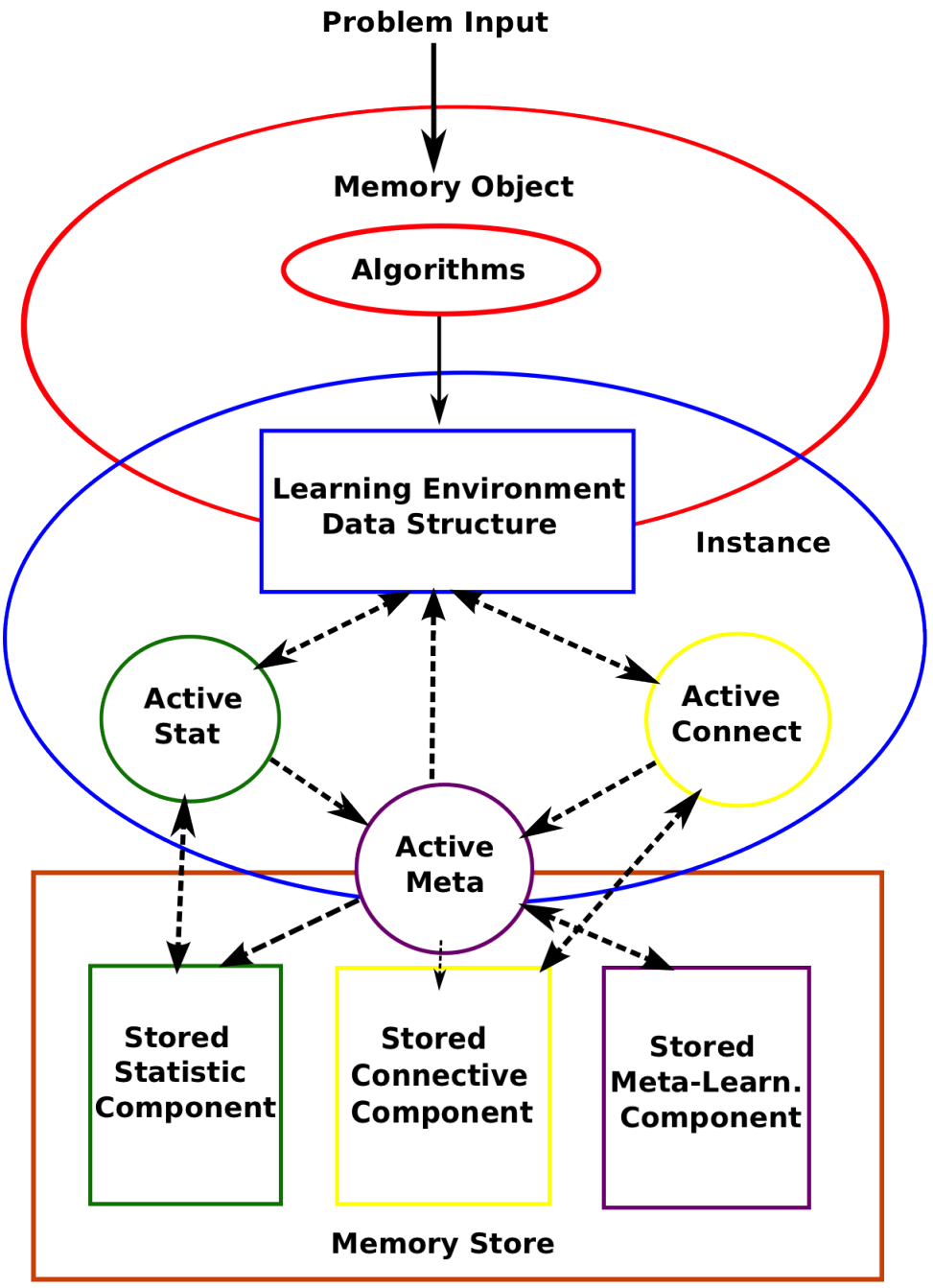

Figure 3.2: Component Overview

environment can be stored for further refinement against additional data sets. This more closely mimics the environment in which humans learn.

2. a training method, which given a set of data, will add the information to memory and attempt to optimize the structure of the data for solving the problem at hand.

3. a results method, which queries the memory components to provide an answer to the question given.

Ideally this object would, dynamically and based on prior problem solving information, design the requirements for the solution. It would then pass the various components enough information to have them intelligently look for the answer. Another ideal characteristic would be for the memory object to handle the parallel processes and synchronize information as it was found. This would allow biases based on unconscious insight or new information coming into focus to change the solution path, allowing a more elastic problem-solving capability. These enhancements would also bring the memory object more in line with Baddeley and Hitch's Central Executive.

3.2.2 Statistical component

The statistical component is an attempt to mimic the unconscious mind, a quick pattern matcher that is the first component of memory to have a possible solution to a problem. Tests have shown [19] that the unconscious

mind is quick to realize patterns, but slow to affect conscious decisions. Ideally it would be continually collecting statistics, running over problems, and attempting to make sense of how disconnected information possibly fits together.

The statistical component is responsible for the following functionality:

1. a method to capture and train information related to words. This method counts and indexes the words in their relation to the target disambiguation. It also captures statistics on each of the targeted word-disambiguation combinations.

2. a voting mechanism, which given a list of words and a target, uses frequency and accuracy statistics of each word to vote on which disambiguation of the target is most likely to be correct.

3. a method to determine which words in a sentence are most likely to lead to a successful completion. This determines the importance of each word based on how often the word has been seen in combination with the target, and how often the word has led to a successful disambiguation.

4. a fast method that, given nothing but the target word, returns the most likely disambiguation based on the most commonly seen word-sense of the target word.

While the model is currently simplistic, this is a necessity with the constraints

of current computer hardware as it relates to the massive parallelism of the human brain and general solution complexity. Ideally, the statistical component or unconscious mind would collect and index every piece of statistical information it could find and, using all non-busy resources, continually refine and apply new information to existing problems. This would provide a much more dynamic best effort guess, which would almost certainly be a better generic problem-solving method.

3.2.3 Connective component

The connective component models long-term memory, a logically organized data structure that attempts to allow a deductive problem solving process. The connective component in combination with the working-memory component allows for "memory" connections to gain strength or to decay, increasing the success rate of determining the deductive path.

The connective component is responsible for the following functionality:

1. a method to get or learn the various words. This method activates, or loads, the long-term memory node of each word from the problem instance, bringing its neighbours into working memory. If a word does not exist in long-term memory, a graph of its neighbours found in the problem set is loaded. The word would be added through the memory object after the completion of the problem.

2. a voting mechanism. This method votes on which is the correct word

sense by searching the memory graph from each word-node. The search acts in a greedy manner, returning the first possible disambiguation found of the target word. The word sense returned with the most results is determined to be the correct word sense.

In addition to the above functionality, the data structure itself allows for feedback from the working memory object. This feedback reflects the success of the currently completed problem, which may put a higher cost on traversing edges that led to an incorrect answer, or reduce the cost on edges that contributed to a successful answer.

The expressiveness of a graph in modelling long term memory does allow for a natural correlation. While the form that long-term memory takes, how it is used in problem solving, and how it works with working memory is fairly ambiguous in any of the models, a number of improvements could be made to bring the memory more in line with the psychological models:

1. dimensionality - Currently the long-term memory object is set up to solve just the disambiguation problem; no other data outside this problem are collected and no additional dimensions related to words are used to make better sense of a problem. Ideally, the neighbours of a word could pivot on information found in one of the additional dimensions of data collected, allowing for better accuracy.

2. unconscious access - As mentioned above, in a parallel environment similar to the brain's, ideally the unconscious working away would affect

the memory graph (slowly), applying information learned in pattern matching to the overall structure of the graph.

3.2.4 Meta-learning component

The meta-learning component is an attempt to bring some of the functionality of the central executive and methodology of problem solving to the current model. It is responsible for various meta-values that are added into the algorithms of the other components. This attempts to allow the current model to better learn how to learn. It adjusts how it solves problems based on trial and error. The tracked values are values that can affect the outcome of a problem, without changing or modifying actual problem data.
Currently tracked values include:

1. important words - the number of words to use when evaluating a given sentence for importance

2. graph depth - the maximum depth to which the voting algorithm should go when attempting to vote using a word

3. graph correctness - the value by which to increment the edge weight between the given node and target node when it is correct

4. graph incorrectness - the value by which to decrement the edge weight between the given node and target node when it is incorrect

5. update frequency - how many solutions should pass between meta algorithm updates

Each value contains the current numerical value, the complete history of changes and the maximum step amount between changes. The algorithm for meta-learning randomly selects a value to change, changes it, and then uses the hypothesis testing to determine if the outcome was worth it. If it was, it changes it again; if it wasn't, it reverses the value and randomly selects another value to modify.

In addition to this, the meta-learning component is also used to track how each of the other two components performs and adjusts their voting confidences based on past performance, ultimately giving the meta-learning component the final say on voting success.

It also puts the working memory information learned in each problem instance into practice, modifying the statistical and graph information contained in the other two components.

Ideally, expanding on the meta-tracking information to include all values, and eventually all methods and general problem solving strategies, would allow the meta learning object to become more like the brain and move from specific problems to general problems. The complexity of this increases quickly, as does the number of resources required (massive parallelism again). The necessary information to track would grow and, more importantly, the number of training examples required to generate good results would be very large. Initially, genetic algorithms would be used just to get off the ground,

but tackling the executive function problems and then the specific component algorithm creation requires something outside of the scope of this thesis.

3.3 Base Models

Now that the system has been outlined at both a high level and a component level, an overview of the decisions that were required will help the reader to better understand the complexity of the model. This section analyses the components against the models, shows their shortcomings and attempts to justify a number of the decisions made.

3.3.1 Baddeley and Hitch

In the initial implementation, the main component taken from Baddeley and Hitch's model was the central executive. The central executive was responsible for scheduling tasks, coordinating the underlying systems, and making sense of the results from the sources. In the final implementation, with the removal of parallelism, the tasks became more static. Although parallelism is essential to modelling the brain and mind, using it to tackle only the one problem presented had minimal gains for the amount of effort required. The functions that made up the original executive were moved into the memory workflow. Methods that coordinated the system's tasks and handled the passing of values and components between the various memory components remained, but were simplified. With the reduced constraints

due to the serialization of tasks and workflow, the scheduling component was removed, as dynamic execution was no longer required.

The looping components of the model were reduced from the four components outlined in Section 2.2.1 into one main looping structure. This looping structure consisted of the train, optimize and guess loop. The training part initializes and loads the long-term memory component and the unconscious mind's statistics engine. The optimize part simulates one cycle through the "learning" loop, like the Baddeley and Hitch model; information is processed, connections are made and removed, and the long-term memory structure becomes more organized. Ideally, just as in the model, this portion of the loop would iterate several times, strengthening the structural learning components. This process being slow in humans, who have the luxury of gathering it over many decades, it is also quite slow for the non-parallelized program (and, even then, work on handling simultaneous reads and writes would be required). Finally, the guess loop uses the metrics gathered and the long-term and statistical information created in the optimize part to find the most likely solution to the query. Another aspect taken from the model is that of influence between components. Where the Baddeley and Hitch model parallelized the input, and handled it in specific memory objects, this project had a single point of input which was then transferred into two models of viewing it, with the memory component and unconscious mind component acting as the central executive.

While the overall structure is similar, it is simplified to fit. The main aspect

of the Baddeley and Hitch model is how the multiple loops and buffers are processed and how they interact with the various memory components. The phonological loop repeatedly rehearses sounds, such as language, to refresh a phonological area of working memory. This can be used with sounds and language themselves, or through reading words and having the phonological loop repeat the words, putting the words in working memory, strengthening each of the words and the connections. While the phonological loop and memory store are at work, the mind can also be processing other components of the story at the same time in the visuospatial sketchpad or the episodic buffer. The visuospatial sketchpad allows visual information to be stored and manipulated, and tasks involving planning, navigation and physical movement to be ordered and manipulated. The episodic buffer is thought to link all the forms of learning (visual, spatial, and verbal) to a chronological timeline. This component is also the most important in linking to long-term memory and making semantic connections. Now if we take the reading example mentioned above and apply both the visuospatial sketchpad and the episodic buffer, we gain a visual memory of the scene in the book, which we are able to mentally move through. We also have an imprint of the sounds of the words and a timeline of how the scene unfolds and a semantic understanding of the words. The three systems working together have a powerful effect of truly understanding the words, the characters and the scene. With the three acting in concert together, information gained in any of the sections is available to the mind as a whole, thus compounding the learning ability

and increasing the likelihood of making long-term and unconscious connections. Baddeley and Hitch specifically stated that these components work in parallel with one another, so with any information passing or issues handled by the central executive, the learning and understanding potential is much more powerful than the serialized one-track solution designed.

3.3.2 Cowan

The main idea used from Cowan's memory model is the unification of long-term and working memory through the use of focus. The memory components have three states; however, in each state, the memory component has the same internal representation. The initial state, which occurs through the creation of an instance, is done when the problem encountered requires a new memory container. This occurs when the word encountered has not been seen before and could be required to solve the current problem. The memory component is then used for the problem instance, and once the instance is completed and the component is found to be useful, the component's state is added to the long-term store. The next state occurs while the memory component is at rest in the long-term store. When the memory component is not required for a problem, the component lies dormant in a long-term store. For the majority of the time, the component is in this state. In this state, it can be written to a file, serialized, and compressed, mimicking the storage condition of the human brain. Finally, in the third state, the stored memory components are required for a problem instance. The components that gain

focus are activated out of the long-term store and are brought into the current problem instance with the information of past experiences available. In the active state, they provide connection to other nodes that could contain required information, and hold useful statistics used in the problem solving algorithm. During the problem instance, interactions useful to solving the problem are imprinted on the active memory nodes; when the problem is completed, the active nodes are returned to the long-term store, with new problem-solving statistics.

One of the limitations of the implementation, in regards to Cowan's model, is the statistical collection of memory types. Cowan's model allows for varying degrees of detail as required by the degree of focus. In the long-term store, each component holds little information, but the connection of all of the components can be seen. With the first degree of focus, only approximately seven elements are "loaded" into memory, but with this, the amount of information and statistics on each of the seven elements are much greater. While the implementation attempted to do this, the information chosen for collection was hand-selected and focused on what was "thought" to be more useful, and was stored at every level (but not necessarily used). In order for this to be more useful, an automated method for statistical collection would have to be used and analyzed by the unconscious mind. It is only then that focusing on a few elements or a single element and allowing a comparison between the deeper statistical information could garner a much greater level of problem solving. Furthermore, in only capturing written text input, the

deepest focus, which loads only a single element but allows for a full view of the element, is missing out on better connections. Whereas humans can add the input from the five senses to a memory component, and in doing so create a five dimensional graph of connections, the implementation deals only in a single dimension, which suffers from a convoluted word space and unsolvable disambiguations.

3.3.3 Ericsson and Kintsch

The important idea in Ericsson and Kintsch's work is the idea of memory linking. The idea is that, in working memory, we are able to hold cues or links to long-term memory components. This allows us to understand complex memories through the use of holding important short-term concepts which can quickly bring the more abundant long-term memories into focus. This model works very well in practice. In the working model, each memory component that is loaded into short term memory is little more than a cue to its place in the long-term memory data structure. The memory component contains some information about itself and a list of its neighbours. The algorithms for readying the working-memory structure for problem solving first attempt to load the memory objects (and therefore cues) that seem most relevant. Then, in the process of solving the problems, each component is analyzed, and if during the process of solving the problem, any of the cues seem overly relevant, the more strongly related neighbours are loaded into memory. This allows the relationship of the memory components seen

during the initial look at the problem to be the starting point. Then, using the complex collected relationships of each memory component's long-term links, it begins traversing the set of both nodes looking for similarities. When the perceived strongest relationship links to a disambiguation that matches the word in question, the connective component's search for an answer is considered complete.

Another important idea in Ericsson and Kintsch's work is that there are different retrieval structures dependent on the nature of the subject matter contained therein. While the working model created uses a combination of the deliberate structure and the text comprehension structure, it is not capable of separating the two methods or of applying the pattern and schema's structure. Separating the two structures currently in use would allow the model to put more importance on links that have a higher likelihood of solving a problem. Instead, the deliberate structure is used on every potential item in each sentence, making each item as important as every other item. It does use text comprehension to remove items that are not relevant, focusing instead on the nouns and verbs of a sentence. To improve the memory structure further, applying the patterns and schema to the solutions where relevant would greatly increase connective memory accuracy. This would require three distinct parsers with three separate learning engines. The sentence parser would act as it currently does and would rate the importance of each word. At the same time, the deliberate memory parser would determine if any of the words or ideas in the sentence link to any

flagged memories, triggering their addition to the solution equation. This portion would require a method to determine if a given memory component has some value that an algorithm would determine to be anomalous enough to warrant a deliberate memory link. Finally, the third parser would be the pattern and schema parser; this would abstract and compare the current sentence and ideas within the sentence to other abstracted memory component collections. This parser would then determine likely outcome memory components or memory areas to narrow the search.

The current model does perform a number of what the parsers described above do; however, a more complex memory component model would be necessary to provide the abstractions needed to determine general problem-solving patterns. Furthermore, developing the algorithms to determine which memory components are more important and should be used as a future indicator of success where possible requires more research.

3.3.4 Unconscious Mind

The unconscious mind component was initially not a planned component. It came from the necessity of having an algorithm that did not require the connective memory component to determine which words in a sentence were important. In researching the unconscious mind, and how humans use their "gut" in problem solving, it was evident that a component that tracked simple statistics was required. The unconscious mind became the basis for determining the best words in a sentence, the statistical memory component

of the overall solution algorithm, and assisted in the meta-learning algorithm. While no definitive models exist for what exactly the unconscious mind does, the Iowa gambling task [11, 12] mentioned in *Blink* [19] adds credibility to an unconscious component of the brain forming a reaction well before it enters conscious thought. It is assumed that some form of statistical analysis is performed and it is of great use in decision making. The idea of the unconscious as a statistical solution engine guided the design. The studies on the use of unconscious processes emphasized the difference between declarative knowledge, which allows one to articulate reasons for choices, and procedural knowledge, which allows repeated complex actions to be performed at an unconscious level. Where declarative memory takes longer to recall, the mental processes that act on it allow more useful relative information to be made available. On the other hand, procedural knowledge can, over time, move from an unconscious understanding to a long-term reflexive understanding. The person is able to know something or properly mimic an action but it is done at a level that requires little, if any, conscious thought. Procedural knowledge is more disconnected from neighbouring memories, and even an attempt at describing the process is often difficult.

The implementation of the unconscious mind in the current model does attempt to add in procedural knowledge. However, unlike true procedural knowledge, the implementation's version has a much more conscious process. Currently, the model relies on a predefined collection of statistical information. This information is then used in a number of predetermined algorithms;

some, like the hypothesis testing algorithm, allow for flexible and adaptable use of information that most do not. The ideal solution would allow for emergent behaviours in both determining what statistics to collect and how to use the statistics to best solve the current and future problems. Another issue that would allow for greater success is having the unconscious mind completely separate from the other components, collecting the information and statistics anonymously. This would allow for both a better collection of information and more up to date information when needed. The unconscious mind could also spend a great deal more of its time making, testing and re-defining multiple hypotheses, without affecting the conscious work loop.
The statistical engine in its current state tracks a number of useful properties including:

- the number of times a word has been seen;
- the number of times the word has been in a successfully disambiguated sentence;
- the number of times the word has been used successfully in the statistical and connective algorithms;
- a list of each word that has been seen by a disambiguation, and the number of times it has been seen; and
- a list of each disambiguation that a word has seen, and the number of times it has been seen.

Chapter 4

Implementation

This chapter deals with the implementation details of the system and related algorithms. This includes specifics on the structure of the input data, the training setup and procedures, the learning and adaptation specifics, and how the testing was performed. Whereas the last chapter discussed design decisions, this chapter deals with how the ideas were implemented in the system. This chapter also discusses the outcomes of the various implementation choices and attempts to further justify the design decisions.

With this section presenting specifics of the test implementation, a restatement of the problem domain is in order. Given a sentence and a target ambiguous word in the sentence, can a system properly disambiguate the word, providing the correct word sense? Therefore, we are trying to disambiguate a word with multiple word senses in a given sentence, where disambiguation is the action of determining which word sense is correct in the given situation.

4.1 Input Data

The input data to the test were taken from the Natural Language Tool Kit (NLTK), an open source python project for use in linguistics and natural language processing research and development[29]. The NLTK is used specifically for its symbolic and statistical natural language processing, which is a good fit for the topic of this thesis. The main use of the NLTK was to provide the Senseval-2 contest data set. The data set is made up of manually tagged sentences made publicly available through the Senseval/Semeval workshops, a set of workshops put on by the Association for Computational Linguistics to promote the semantic analysis of text. These sentences are tagged in a reproducible manner. Each sentence was added with the rule that human taggers must agree on the sense of each word at least 90% of the time, requiring at least two taggers to determine the validity of each word. The Senseval-2 data set consists of 15225 sentences, containing an average of 27 components. The sentences were organized in a strict XML format, with each record of both the training and testing dataset containing the following elements:

- word- the targeted word and its abbreviated part of speech. In the Senseval-2 data set there are 4 words to be disambiguated: “hard”, “interest”, “line” and “serve”. Each word contains a possible ten to thirty-six word senses; of these, the dataset included four to six candidate word senses per word. An example of this category is simply

"Hard-a", with the "-a" being representative of its part of speech (in this case, an adverb)

- context- the sentence in which the targeted word is found. This consists of a set of tuples comprising each component of the sentence; each tuple contains a stemmed word or component, and its part-of-speech tag. For instance:

  ```
  [('someone', 'NN'), ('has', 'VBZ'), ('to', 'TO'),
  ('stop', 'VB'), ('him', 'PRP'), ('.', '.')]
  ```

- senses- The word senses of the targeted word, taken from the WordNet 1.7 sense inventory. An example is "HARD1", which can be cross-referenced for its definition.

In addition to the data set, the NLTK also provided a mechanism for stemming and lemmatizing each word, as well as providing a list of commonly used stop words. This allows each sentence to be more compact and concise, which greatly reduced the overall complexity of the graph-based memory component.

4.2 Setup

The data were set up using five-fold cross validation, in which four subsets of the data set were used for training and the fifth was used for testing. This

was then repeated five times, allowing each of the data sets to be the test data, and having every combination of sections used for the training data. This method differed from the method used at the Senseval-2 competition, which only supplied training data and kept the testing data hidden, but allows for an acceptable comparison. The Senseval-2 competition gave the researchers a full training set, but the testing data were withheld until after the competition was completed.

4.3 Training

The training component of the system provided a non-trivial challenge. To create and synthesize the learning process in humans, a two-phase approach to training was required. The first phase consisted of the initial setup of both the statistical component and the connective component. The connective component required its "memory connections" to be added. This was the process of adding weighted edges between words that were found in the same sentence, where the edge weight was directly proportional to the frequency of occurrences between the two words. Essentially this built the unoptimized long-term memory graph, with a simple estimate for the edge weight, the number of times the two words appeared together in a sentence. The statistical component collected word-based frequency information that would be immediately relevant for problem solving. This first phase ran over the entire set of training data.

Once this initial processing was done, a second pass of the training material was required. This pass could be considered the learning phase and has the goal of optimizing the edge weights of the memory graph, verifying the statistical component's accuracy, and balancing the weights behind the meta-learning component's decision making process. To optimize the memory graph, the training data were used again. With the structure of the graph complete, the disambiguation was attempted by using various paths through the graph. If the paths were successful at disambiguating the word, the edge weights were increased; otherwise they were decreased. The statistical component was also checking its accuracy throughout the second pass of the training data. At the end of the second pass, both components had an accuracy associated with them. These accuracies were then passed into the meta-learning component, which used them to help determine how to select the best outcome, given the results of the statistical and graph-based components. Furthermore, using the statistical component's word frequency information, the meta-learning component could determine which words in a given sentence would most likely end in a successful word-sense disambiguation.

In order to select the best words from the sentence, the following three factors, based on both the word in question and the word to be disambiguated (target word), had to be considered:

- Accuracy: the probability the target word would be accurately disambiguated.

- Completion rate: the number of times the word has been involved in a successful disambiguation.
- Rarity: the frequency with which the word occurs in the training.

In the end, the algorithm used is shown below, where the accuracy function is defined as the number of successful disambiguations in which W has been involved.

input : T (the target word) and W (a word being compared to T)
output: A value of how "good" W is in relation to T

$N \longleftarrow 0$
$D \longleftarrow 0$
foreach *t, a word sense of T,* **do**
    $N \longleftarrow N$ + number of times t has been in a sentence with W
    $D \longleftarrow D+$ total number of words t has seen
end
return $(N/D)*$ `accuracy(`W`)`

Algorithm 1: best word algorithm

Finally, once the best words were known, the number of words to choose per sentence was decided by the trial and error method in the meta-learning component.

4.3.1 Graph-based adaptation and query

The method by which graph-based adaptation and queries took place was through the use of collected meta-data and instance-based feedback. In the initial graph setup, the words in each sentence were used as nodes. The

number of times two words were seen together was the initial weight of the edge between them. During the second pass of the training data and during the test phase, the graph was used to discover the best word sense. It did this by beginning a search on each node related to the chosen best words of a sentence. The algorithm first looked to see if a disambiguation of the target word existed in its neighbours in an edge-weight-descending order. If one did not, it traversed the most highly-weighted edge and looked at the new node in a similar manner. When the search ended, each node had a candidate word sense, and the word sense with the greatest tally was returned. The algorithm is more formally presented as Algorithm 2.

Algorithm 2 presents how the graph component is used to disambiguate a word; the adaptation component requires a bit more background. The adaptation phase is the process of modifying the edge weights of the graph to better the outcome of subsequent graph queries or searches. The graph adaptation algorithm is performed after the problem instance has completed and the correct disambiguation is known. During this phase, the outcomes are:

- The graph component correctly determines the word sense. In this case, the edge weights along the path each selected word traversed has its value increased.

- The graph component incorrectly determines the word sense. In this case, each edge weight along the path that led to the wrong outcome

```
input  : The target word T, a list of word nodes related to a
         sentence L, a max depth X
output: A word sense of T
List Y
foreach n in L do
    if n.count +1 > X then
        Y ⟵ ∅
        break
    end
    sortedge(n, Descending)
    foreach neighbour s of n do
        if disambiguationOf (T, s) then
            Y ⟵ s
            break
        end
    end
    u ⟵ neighbour(n)
    u.count ⟵ u.count +1
    L ⟵ u
end
return getlargestcardinalelement(L)
```

Algorithm 2: graph search

would have its value decreased. This value was often significantly larger than the increased value for successful outcomes.

In keeping with the human memory modeling, the values used to increase and decrease the edge weights were set using the trial and error method in the meta-learning component.

After completing the second run of the training data set, the words were known and the edge weights of the graph had been trained enough for use. Although it has not been confirmed if additional runs of the graph adaptation phase would be beneficial, the length of time this phase takes and the increased likelihood of overtraining were the deciding factors in using one run.

4.3.2 Statistical adaptation

In keeping with the unconscious mind, the statistical method was designed specifically to be a quick, relatively simple calculation with decent results. a good approach for this, and one that could be related to the actual process, was Bayesian statistics. After each learning instance, the statistical information contained therein was added to the previously collected information. This information was immediately available for use in calculating probabilities for the next learning instance. The first part of the learning phase, the statistical component, collected the number of times words were seen together, attributing each sentence and each word in the sentence to its proper

word sense. The second phase, the optimization component, attempted to select the correct disambiguation based on prior knowledge, and would again add the information to its statistical collection. In the initial system design, the selection algorithm used a wider range of variables to determine which word sense was most likely; this, however, proved to be a poor choice, as a simple most-frequently-selected heuristic worked much better. This led to a revised version of the system, using the very simple "most frequently selected" algorithm. This algorithm uses a frequency table to keep track of all of the different word senses of a single word; when asked to disambiguate a given word, the algorithm simply returned the word sense that has the highest number of occurrences.

4.3.3 Meta-data collection and adaptation

During the second run of the training phase, the meta-data collection and adaptation are completed. This phase consists of two important parts:

- answer biasing- This phase adds a bias to the answers provided by the two other components in accordance with past accuracy. This uses information collected in previously answered instances to add weights to each of the answers provided by the components. It uses these weights and the confidence each component has in its answer to choose the final word sense.

- adaptive processing- This component monitors, tests and adapts the

pre-defined meta-values. After each problem instance, a check is done to see if processing is required. If processing is required, a meta-value is chosen to go through a trial and error testing phase. This phase involves modifying the value by a pre-defined "step" value. Training or testing continues with this value set; after a number of instances, the processing phase is triggered again. The results from this testing phase are compared against the captured performance statistics. If the test returns a significant improvement, the new value is set as default and the meta-value is again selected for testing. If the test returns a failure, the value is reset and another is randomly chosen. Finally, if the value sees a moderate improvement, the value is set and a random meta-value selected for testing.

The meta-variables on which these tests were performed consist of two groups. The first group is the "desirably controlled quality" group. This group consists of values that are not directly related to accuracy, but that likely have an effect on the results. The main reason these were added was to have the system optimize itself throughout the learning phase.
Desirably controlled quality:

- important words - the number of words selected from a sentence used to determine the word sense. This one was selected in hopes that, as the algorithm became more efficient, or during the initial learning phase, the number of words required could shrink or grow. This would

allow the algorithm to determine when it should use more words and run more slowly, while maintaining or increasing accuracy, or when it should use fewer words and run faster without affecting accuracy.

- iterations between considerations - This represents the number of test instances between the meta-data processing phases. It was expected that this value would be the last value selected once the other values found an equilibrium, as any results other than this would lower accuracy. Once the equilibrium was found, this value would increase, optimizing the speed of testing.

- graph depth - This is the maximum depth to which the graph algorithm would go before guessing. As with important words, it was thought that it would find the best compromise for accuracy initially, and as the graph became more accurate, it would be increased to allow for faster runtimes.

The second group is the "learning optimization" group. This group consists of values that are too hard to set statically, and would likely have a moving optimal target. The main reason these were added was to conform to human memory models, as well as the inability to determine an optimal value.
Inflation limiting:

- Positive Outcome - This was the value gained for each edge used in determining a correct word sense disambiguation.

- Negative Outcome - This was the value removed for each edge used in determining an incorrect word sense disambiguation.

These groups and the meta-learning component allowed for a great chance to add accuracy in modelling the biological and cognitive aspect of memory and learning, with a likely increase in overall accuracy and efficiency. Along with this, there was also a number of possible benefits such as a scaling reward and punishment system, and an algorithm that was slower and more deliberate in its searching when its accuracy was suspect, and faster when the learning component was performing efficiently. It also had the added effect of eliminating the guess work in initial assignment of these values.

Chapter 5

Results

In this section, the results of the human memory model will be presented. Interpretation of the results and commentary on both what worked and what did not work will be discussed. An overview of the approach and the novelty of its implementation will be looked at. Finally, the relative success of the work will be outlined.

The comparisons made in the chapter are broken down into quantitative results and discussion sections. The quantitative results section compares the current system's performance to systems that had competed in the Senseval-2 contest, including the baseline algorithm. The discussion section generalizes the trends discovered in the data from the quantitative section. This section analyses pieces of data that seem interesting or inconsistent, hypothesising the reasons why, and adding insight into potential issues.

5.1 Quantitative results

5-fold cross validation results		
test instances	**accuracy**	**std. dev.**
3045	0.5997	0.0284
3044	0.5957	0.0279
3045	0.5934	0.0287
3045	0.6020	0.0286
3045	0.5915	0.0284
overall accuracy: 0.5967		

Table 5.1: Results Table

The quantitative results focus on accuracy, or the percentage of the time in which the correct word sense is being chosen. In Table 5.1, we see the results of each of the 5-fold cross validation test instances run on the current system, with each test instance attempting to disambiguate one of four possible words, and with each word containing at most six unique senses. With an overall average of 59.7% this test fell just within a standard deviation of the baseline result of 56.9%. The baseline algorithm for the Senseval-2 contest performs surprisingly well given that it simply chooses the word sense that occurs the most frequently. Table 5.2 presents all competitors in the Senseval-2 competition as it relates to the word sense disambiguation problem. While none of the other models use a human-like memory approach, they did use a range of artificial intelligence techniques including: memory-based learning [16, 22, 40], instance based learning [30], decision lists [41], ensemble methods [18], kernelization [20], as well as various knowledge hy-

brid methods [34]. The results are sorted by recall where recall indicates the percentage of positive cases that were correctly identified and precision is the percentage of positive predictions that were correct. The table also shows how well the baseline did in relation to other systems. Added to the table (in bold) are the results from the current learning method and a self-created implementation of the baseline test. The self-created implementation of the baseline test falls within 0.4% of the competition's baseline test, validating both the data sets and the testing approach used.

The information in the regression analysis of training accuracy (Figure 5.1) holds the results of all instances over the entire 5-fold cross validation process, with each dot representing an accuracy calculation during each of the training runs. The x-axis represents the number of training instances processed up to that point (x1000), and the y-axis represents the accuracy. Each fold consists of approximately 12000 training instances, and after each fold, all previous learning data structures are reset, the only hold-over being the changes to the meta-variables. The first trend is shown with the solid line, showing a steady improvement over the entire test. The 2.5% gain on the average accuracy over the 5 folds can be attributed to the meta-learning component and the preserved meta-variables mentioned earlier. The other important trend is found in looking at the general improvement of values over one test fold. To make this clearer, the regression analysis of one fold can be seen in Figure 5.2. The important thing to note when looking at the training data set is that the statistical component was pre-optimized during the training

English all words - fine-grained scoring			
precision	**recall**	**attempted**	**system**
0.690	0.690	100%	SMUaw
0.636	0.636	100%	CNTS-Antwerp
0.618	0.618	100%	Sinequa-LIA - HMM
0.597	**0.597**	**100%**	**Current technique**
0.573	**0.573**	**100%**	**self-created baseline implementation**
0.575	0.569	98.91%	UNED - AW-U2
0.569	0.569	100%	Baseline
0.556	0.550	98.908%	UNED - AW-U
0.475	0.454	95.552%	UCLA - gchao2
0.474	0.453	95.552%	UCLA - gchao3
0.416	0.451	98.5%	CL Research - DIMAP
0.451	0.451	100%	CL Research - DIMAP (R)
0.500	0.449	89.729%	UCLA - gchao
0.360	0.360	99.96%	Universiti Sains Malaysia 2
0.748	0.357	47.756%	IRST
0.345	0.338	97.897%	Universiti Sains Malaysia 1
0.336	0.336	99.96%	Universiti Sains Malaysia 3
0.572	0.291	50.789%	BCU - ehu-dlist-all
0.440	0.200	45.37%	Sheffield
0.566	0.169	29.883%	Sussex - sel-ospd
0.545	0.169	31.055%	Sussex - sel-ospd-ana
0.598	0.140	23.332%	Sussex - sel
0.328	0.038	11.646%	IIT 2
0.294	0.034	11.646%	IIT 3
0.287	0.033	11.646%	IIT 1

Table 5.2: Senseval 2 Results, with the current technique and self-baseline implementation added

bootstrap phase. Therefore, since the statistical component did not change over the life of the test, the improvement can be attributed to the connective component, as it is still being optimized. While the meta-learning changes are cumulative over the whole graph, the nearly 5% increase in accuracy over the course of a single fold's graph can be credited to the training of the connective component. Furthermore, as the accuracy increases over the entire test at a steady rate related to the meta-learning changes, and the rate of improvement during a test fold is more accelerated, it can be seen that both components independently improve accuracy. From these assumptions, we can then posit that accuracy within one fold is improved in the majority by optimization in the connective component. Overall accuracy improvement, or perhaps refinement over the course of retraining the learning components, can be attributed to the meta-variables and the training done by the meta-learning component. Furthermore, the results show conclusively that the accuracy and number of instances are positively correlated; therefore, adding the meta-learning components and the connective components have added value.

Tables 5.3 and 5.4 show statistics about the regression analysis tests on Figures 5.1 and 5.2. The tables attempt to determine a formula for the accuracy given the number of instances. That formula is $y = \alpha + \beta x + \epsilon$; in the case of Figure 5.1, we see that starting at a baseline of 60.5%, each instance increases the accuracy by $(3.757 \pm 1.614) * 10^{-4}$. The confidence value of α shows that the confidence interval will incorporate 99% of all instances'

results, whereas β's confidence interval incorporates 95% of all instances' results. The t-statistic measures how reasonable the estimation is; the closer to zero, the more reasonable. The extremely large value in both tables' α variable shows that the values are serially correlated. This makes sense as this value is highly dependent on its previous values; in this case, estimation errors in earlier instance runs tend to be correlated with estimation errors in newer runs. This is expected when collecting data repeatedly across time and will not affect the consistency or lack of bias in estimates; it does usually give a more precise confidence interval than it actually is. The β variable t-statistic being greater than 2 means it is significant and has a high impact on the accuracy. This is also expected, as the increase in instances being trained against should increase accuracy. The p-values of all the variables are less then the remainder confidence percentage, which indicates that the results shown are non-random and that the variables have an effect on the data set.

Regression analysis statistics (entire training)				
model	$y = \alpha + \beta x + \epsilon$ where y is accuracy and x is instances			
	estimate	**confidence**	**t-statistic**	**p-value**
α	0.6058 ± 0.0047	$> 99\%$	128	$7.3 * 10^{-4}$
β	$(3.757 \pm 1.614) * 10^{-4}$	$> 95\%$	2.3	0.024

Table 5.3: Statistical values of regression analysis on entire training data

To validate how well the memory model setup worked, we will again look to the training data leading up to the above test results. In the training results table (Table 5.5), we see the total results for all of the training of the cross-

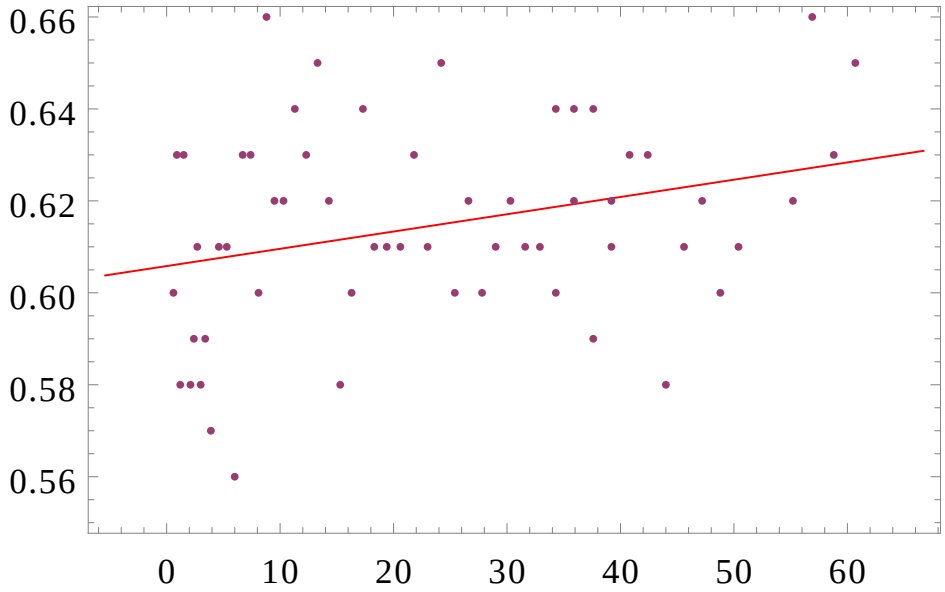

Figure 5.1: Regression analysis of training accuracy (x Instances x1000, y Accuracy)

Regression analysis statistics (one training fold)				
model	$y = \alpha + \beta x + \epsilon$ where y is accuracy and x is instances			
	estimate	**confidence**	**t-statistic**	**p-value**
α	0.5895 ± 0.0091	$> 99\%$	65	$9.8 * 10^{-24}$
β	$(3.498 \pm 1.428) * 10^{-6}$	$> 95\%$	2.4	0.024

Table 5.4: Statistical values of regression analysis over one fold

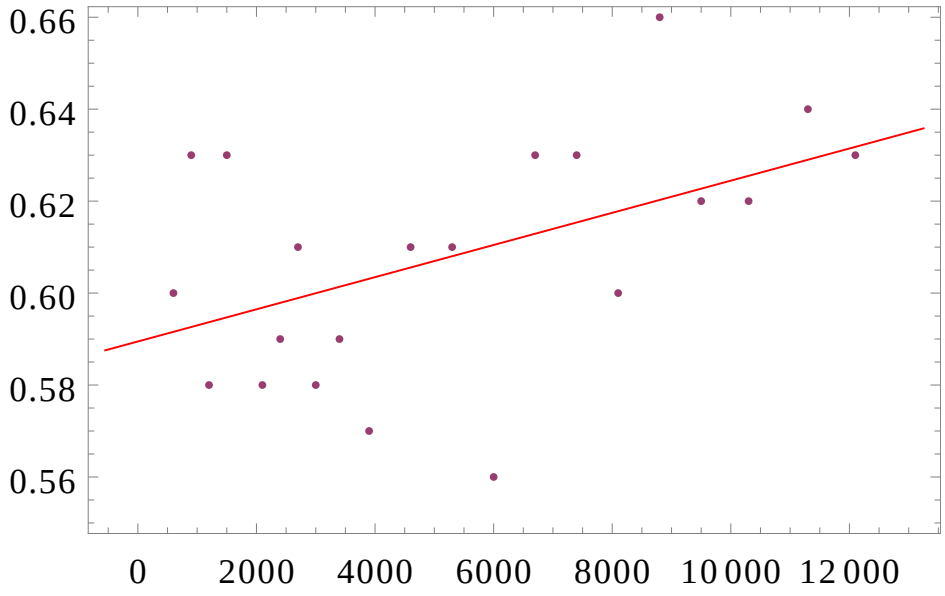

Figure 5.2: Regression analysis of training accuracy over one fold (x Instances, y Accuracy)

validation runs. In total 60700 training instances were run, of which 37587 (61.9%) were successful. The statistical component, which represents the snap judgement of the unconscious mind was correct 59.6% of the time, which made up 96.2% of the correct results. The connective component, which represented the deductive mind, was correct 37.3% of the time and made up 62.5% of the correct results. This is not a completely fair comparison, as the statistical component was pre-trained during the test bootstrap phase, and the connective component was being trained throughout the training run. It is obvious that there is a large overlap between the accuracy of the two components; however, even from Table 5.5, we can make out that the statistical component does not dominate the graph component.

Training results				
total inst.	**correct**	**percentage**	**connect. correct**	**stats correct**
60700	37587	61.9%	22618 (37.3%)	36166 (59.6%)

Table 5.5: Overview of training results.

In Table 5.6, the accuracy section is the outcome of the algorithm used to determine disambiguation. The statistical and connective accuracy is how often the individual components chose the correct disambiguation. The exclusive statistic and connective is how often the components are correct while the other component was incorrect. Finally, the optimal accuracy is the outcome that could be attained provided the meta-learning component performed optimally and chose the correct value from the components every time a correct value was presented. This table and the information presented

earlier clearly show four things:

- Both the statistical and connective components can add value to the outcome. That is to say, the two components can come to a higher accuracy together than they could by themselves.
- The meta-learning component is unfairly biased towards the statistical component.
- There is something drastically increasing the accuracy of the connective component in the training results.
- With an optimal accuracy approximately 25% higher than the testing accuracy, there is room for optimization and improvement.

Clearly in the training data, and somewhat in the test data, we see a benefit to the two input systems. Although the results are slightly higher than that of the baseline test, there is potential for significant improvement. The training results point to a deficiency in the way the meta-learning component weighs and chooses the target disambiguation. With the connective component improving its accuracy beyond the statistical component, the training accuracy should have a higher rate of selection. We see in Table 5.7 that this is not the case. This leads to the second point, that the meta-learning component may have a more than human bias. In the most extreme case, it chose to go with the statistical component 99.01% of the time despite the connective component exclusively selecting the correct value 12.12% of the

time, and having an accuracy of 39.44%. Finally, the large difference in the connective accuracy between the training and testing results is due to the differences in learning behaviour. The training results allow the connective data and the meta-learning component to modify its learning behaviour per instance, whereas the testing results have both of these locked in. Quicker feedback results in better accuracy, and was done to simulate human testing feedback. However, to act as a Senseval-2 test entry, this mechanism was not used during the testing, as none of the other systems had access to immediate feedback. This trait appears to positively effect the connective component's accuracy, but has very little effect on the meta-learning component.

Training results					
acc.	**stat. acc.**	**excl. stat.**	**connect. acc.**	**excl. conn.**	**opt. acc.**
60.84%	59.54%	26.20%	55.02%	21.67%	81.22%
60.30%	59.62%	26.11%	57.45%	23.94%	83.56%
60.90%	59.68%	24.54%	59.51%	24.37%	84.05%
60.60%	59.45%	25.06%	60.23%	25.88%	85.33%
60.75%	59.72%	21.33%	65.02%	26.63%	86.34%
Testing results					
acc.	**stat. acc.**	**excl. stat.**	**connect. acc.**	**excl. conn.**	**opt. acc.**
59.90%	59.77%	31.43%	39.34%	11.00%	70.77%
59.58%	59.55%	34.83%	36.05%	11.34%	70.88%
59.30%	59.30%	32.26%	38.70%	11.66%	70.96%
60.20%	60.23%	32.91%	39.44%	12.12%	72.35%
59.15%	59.15%	27.75%	43.58%	12.18%	71.33%

Table 5.6: Various accuracy comparisons between components

Training results			
stat. chosen	**conn. chosen**	**stat acc.**	**conn. acc.**
94.23%	5.77%	59.54%	55.02%
93.53%	6.47%	59.62%	57.45%
94.26%	5.74%	59.68%	59.51%
94.43%	5.57%	59.45%	60.23%
93.01%	6.99%	59.72%	65.02%
Testing results			
stat. chosen	**conn. chosen**	**stat acc.**	**conn. acc.**
97.67%	2.33%	59.77%	39.34%
98.78%	1.22%	59.55%	36.05%
98.49%	1.51%	59.30%	38.70%
99.01%	0.99%	60.23%	39.44%
98.85%	1.14%	59.15%	43.58%

Table 5.7: Component bias

5.2 Discussion

The results are competitive with systems from the Senseval 2 test; the system performs respectfully if a little low given the 10 years that have passed since Senseval 2 However, the novel approach of applying a human-like memory model to this test has given some insight into its potential as a learning algorithm, the challenges this method faces, and some of the understanding required to improve its likeness to human memory and overall accuracy.

The outcome and results present a plausible system which uses many features of the human memory models discussed in earlier chapters. Despite an error related to bias, the central executive, or meta-learning component performed particularily well, adding accuracy through a better understanding of how the meta-variables interacted. In Table 5.6, we see in the training results the

connective component grow from 55% to 65% accuracy over the course of the five training runs. The test results are not as extreme but a steady increase in accuracy is seen. The ability to use a guess or gut feeling (the statistical component) to have a decent chance at being correct, while learning some more of the deeper connections over time between words used and disambiguations, is very human-like. When required to give an answer, the system used the most seen candidate, while the more complex cases and patterns were being learned by the "long term" memory component. Although the meta-learning component did not allow the two components to be completely successful, as it was overly biased and flawed in its decision making, it did do some things well. It improved or better optimized the system's ability to learn, by meta-optimizing the learning parameters. Like a human, repeated problem solving approaches will not only improve the area being solved, it will also sharpen the problem solving toolset. This optimization system had the meta-learning component make a hypothesis on what would work well, test its hypothesis over a training run, and validate whether the change it proposed had a positive effect on the system. This allowed for small incremental gains to carry over to successive runs, allowing each initial state to spend less time bootstrapping and more time increasing accuracy. On the other hand, the meta-learning components algorithm used to evaluate which of the two components to use had faults. In looking at the designed algorithm, and the outcome of its training and testing, it is likely that the perfect memory of the algorithm caused issues. The initial state of the statistical

component having a decent accuracy and the necessity for the connective component to bootstrap itself before having a comparative accuracy allowed the algorithm to put too much weight on the statistic accuracy. This bias is useful initially, as it keeps the accuracy high, but fails to adjust over the thousands of iterations.

In this work, the memory system uses three components in an attempt to simulate not only the different components of the brain, but the multiple senses humans use to better store information. While the five human senses were abstracted down to two components with different views on the same data, these components had a dual purpose of also simulating the different types of memory. Using differing views on the data provided greater overall accuracy. Although simulating additional senses from the data provided by the Senseval 2 data sets would be difficult, the benefits of the additional information would likely both increase accuracy and better simulate human memory.

Chapter 6

Conclusion

This thesis has shown it is possible to use recent ideas from disciplines such as biology, social science, psychology, and cognitive science to illuminate new research areas within the artificial intelligence field. Specifically, looking at how humans learn, acquire memories, and solve problems provides a general problem-solving framework. This framework could lead to improvements in current methods and a better understanding of artificial intelligence problems, while increasing our understanding of the cognitive aspects of human problem solving.

One area that seemed like it could benefit from a human problem solving framework is the word sense disambiguation problem from natural language processing. Using the idea that human learning is tied to human memory, three popular human memory models from cognitive science were presented; these models were then unified and adapted into an algorithm. An additional

component from psychology, the unconscious mind, was added to fill the need for bootstrapping initial instances and adding a fast and semi-accurate method to get answers. The implementation of the human memory model showed a modest improvement over a baseline algorithm, and performed competitively against the top four programs from the Senseval 2 contest. Analysis on the results had shown a number of inefficient areas within the implementation, with a potential increase in accuracy of up to 25%.

With the constraints of mimicking a biological model and having the system perform competitively, a balance was necessary that would respectably showcase both goals, thereby opening alternate paths to better artificial intelligence systems. Using human memory models, human-like learning and human psychological constructs, an effective and novel approach to word sense disambiguation was created. While further research in each area could improve the generic human intelligence framework, the application of this to other problems, and the improvement of the accuracy of the technique, researchers must focus on what is more important. While quick gains can be found in improving the system's accuracy, the potential for ground-breaking innovation stems from improving the biological systems and adapting a general intelligence framework to more problems, more inputs, and more data.

Researchers interested in word sense disambiguation and natural language processing could find the graph theoretic algorithm for word proximity useful. This algorithm, working in conjunction with a feedback-based edge-weighting algorithm, should be adaptable to other problems. With more research in

the meta-learning component, this adaptability could be automated and optimized to the specific problem space. Researchers interested in more realistic biological modelling would find the method in which the differing algorithms connect interesting. They also might be interested in the method used in moving working memory into long-term memory. With the adaptation of text-based memory constructs as a guide to memory creation, and the work of Ericsson and Kintsch on memory components, the modification of the connective component to use more than one "sense" would go a long way in creating a unified human-like memory system. Anyone interested in general artificial intelligence or meta-algorithmics would find the simple yet effective meta-learning system interesting. The results show both the success of the system in determining correct rewards and punishments, and the negative impact an incorrect bias can cause. Ultimately, using provably good meta-heuristics should to be able to improve any artificial intelligence algorithm that requires rewards and punishments. Finally, researchers from cognitive science and psychology may be interested in the interpretation of the various areas of the research used, as adapting the system to perform well in one task in a computer program may provide insight into some aspect of cognition.

6.1 Future work

As this was an initial foray into a novel approach to artificial intelligence, memory, and learning, the results show a number of areas that would be inter-

esting candidates for further research. If the goal of the researcher is to get a feeling for the full-potential accuracy of the human memory model, targeted work in the area of the meta-learning component, specifically the decision making and weighting mechanism, would potentially be low-hanging fruit for increasing accuracy. Another area within accuracy optimization would be the replacement of the static statistical learning algorithm with a more dynamic algorithm; it is likely that there are algorithms with properties that would better fit with the connective components algorithm. Finally, investigating how graph-theoretical concepts could be applied to the connective component to increase its accuracy or quicken its optimization time could lead to good results.

The other option for a researcher is to improve the algorithm to better model human memory. This could include separating the senses from the memory components and handling the increased complexity that would come from this. The majority of this work would be in logically organizing the sense information, and allowing the components to interact within the problem space in working memory and filter back to the long-term memory. A further step would be to decouple the meta-learning component from the problem-solving algorithm, allowing the optimizations and adjustments to occur at any time within a problem, opening additional solution paths. Another interesting angle would be devising additional senses as inputs; this would include both how to simulate new senses on the rather flat data sets, how to incorporate them into the system (specifically how to store them in memory), and how

to update the meta-learning component to handle a new source of information. Perhaps the most interesting and difficult area would be to convert the meta-learning component to use unsupervised learning techniques to find and exploit meta-parameters within the system. This would require an algorithm to learn what items within the system can be parameterized and then through the hypothesis model, to test, compare, and choose the best values to optimize the learning algorithms.

Bibliography

[1] *Semeval-2 evaluation exercise*, `http://semeval2.fbk.eu/semeval2.php`, 2008, [Online; Accessed Dec. 10, 2010].

[2] James Anderson, Paul Allopenna, Gerald Guralnik, David Sheinberg, John Santini, Socrates Dimitriadis, Benjamin Machta, and Brian Merritt, *Programming a parallel computer: The ersatz brain project*, Challenges for Computational Intelligence (Wlodzislaw Duch and Jacek Mandziuk, eds.), Studies in Computational Intelligence, vol. 63, Springer Berlin / Heidelberg, 2007, pp. 61–98.

[3] John R. Anderson, *Language, memory and thought*, Erlbaum, Hillsdale, NJ, 1976.

[4] ———, *Rules of the mind*, Erlbaum, Hillsdale, NJ, 1993.

[5] J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, *An integrated theory of the mind*, Psychological Review **111** (2004), no. 4, 1036–1060.

[6] A Baddeley, *Working memory*, Science **255** (1992), no. 5044, 556–559.

[7] Alan Baddeley, *The episodic buffer: a new component of working memory?*, Trends in Cognitive Sciences **4** (2000), no. 11, 417 – 423.

[8] Yehoshua Bar-Hillel, *The present status of automatic translation of languages*, Advances in Computers, vol. 1, Elsevier, 1960, pp. 91 – 163.

[9] J.A. Bargh and E. Morsella, *The unconscious mind.*, Perspect Psychol Sci **3** (2008), no. 1, 73–79.

[10] John A. Bargh, *What have we been priming all these years? On the development, mechanisms, and ecology of nonconscious social behavior*, European Journal of Social Psychology **36** (2006), no. 2, 147–168.

[11] A. Bechara, A. R. Damasio, H. Damasio, and S. W. Anderson, *Insensitivity to future consequences following damage to human prefrontal cortex.*, Cognition **50** (1994), no. 1-3, 7–15.

[12] Antoine Bechara, Hanna Damasio, Daniel Tranel, and Antonio R. Damasio, *Deciding advantageously before knowing the advantageous strategy*, Science **275** (1997), no. 5304, 1293–1295.

[13] Bruce G. Buchanan, *Timeline: A brief history of artificial intelligence*, http://www.aaai.org/AITopics/BriefHistory, May 2012.

[14] N Cowan, *Activation, attention, and short-term memory.*, Mem Cognit **21** (1993), no. 2, 162–7.

[15] N. Cowan, *Attention and memory: An integrated framework*, Oxford Psychology Series, no. 26, Oxford University Press, New York, 1995.

[16] Bart Decadt, Veronique Hoste, Walter Daelemans, and Antal Van Den Bosch, *GAMBL, genetic algorithm optimization of memory-based WSD*, In Proceedings of ACL/SIGLEX Senseval-3, 2004, pp. 108–112.

[17] K. A. Ericsson and W. Kintsch, *Long-term working memory.*, Psychological review **102** (1995), no. 2, 211–245.

[18] Radu Florian, Silviu Cucerzan, Charles Schafer, and David Yarowsky, *Combining classifiers for word sense disambiguation*, Nat. Lang. Eng. **8** (2002), no. 4, 327–341.

[19] Malcolm Gladwell, *Blink : The power of thinking without thinking*, Little, Brown, January 2005.

[20] Alfio Gliozzo, Claudio Giuliano, and Carlo Strapparava, *Domain kernels for word sense disambiguation*, ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (Morristown, NJ, USA), Association for Computational Linguistics, 2005, pp. 403–410.

[21] Anthony G. Greenwald, Eric S. Schuh, and Mark R. Klinger, *Activation by marginally perceptible ("subliminal") stimuli: Dissociation of unconscious from conscious cognition*, Journal of Experimental Psychology: General **124** (1995), 22–42.

[22] V. Hoste, I. Hendrickx, W. Daelemans, and A. Van Den Bosch, *Parameter optimization for machine-learning of word sense disambiguation*, Nat. Lang. Eng. **8** (2002), no. 4, 311–325.

[23] W. John Hutchins, *Machine translation*, Encyclopedia of Computer Science, John Wiley and Sons Ltd., Chichester, UK, pp. 1059–1066.

[24] Nancy Ide and Jean Véronis, *Introduction to the special issue on word sense disambiguation: the state of the art*, Comput. Linguist. **24** (1998), no. 1, 2–40.

[25] John E. Laird, Allen Newell, and Paul S. Rosenbloom, *Soar: an architecture for general intelligence*, Artif. Intell. **33** (1987), no. 1, 1–64.

[26] J. F. Lehman, J. E. Laird, and P. E. Rosenbloom, *A gentle Introduction to Soar: 2006 update*, 2006.

[27] E. D. Liddy, E. Hovy, J. Lin, J. Prager, D. Radev, L. Vanderwende, and R. Weischedel, *Natural language processing*, Encyclopedia of Library and Information Science (2003), 2126–2136.

[28] Elizabeth F. Loftus and M. R. Klinger, *Is the unconscious smart or dumb?*, American Psychologist **47** (1992), 761–65.

[29] Edward Loper and Steven Bird, *Nltk: the natural language toolkit*, Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics

- Volume 1 (Stroudsburg, PA, USA), ETMTNLP '02, Association for Computational Linguistics, 2002, pp. 63–70.

[30] Rada F. Mihalcea, *Word sense disambiguation with pattern learning and automatic feature selection*, Nat. Lang. Eng. **8** (2002), no. 4, 343–358.

[31] Roberto Navigli, *Word sense disambiguation: A survey*, ACM Comput. Surv. **41** (2009), no. 2, 1–69.

[32] Klaus Oberauer, *Access to information in working memory: Exploring the focus of attention*, In, 2002, pp. 411–421.

[33] David Opitz and Richard Maclin, *Popular ensemble methods: An empirical study*, Journal of Artificial Intelligence Research **11** (2011), 169–198.

[34] Hawkins P. and Nettleton D., *Large scale WSD using learning applied to SENSEVAL*, Computers and the Humanities **34** (April 2000), 135–140(6).

[35] Don Perlis, *To BICA and beyond: Rah-rah-rah! - or how biology and anomalies together contribute to flexible cognition*, Papers from the 2008 AAAI Fall Symposium (Menlo Park, California), AAAI Fall Symposium, AAAI, The AAAI Press, 2008, pp. 141–145.

[36] S. J. Russell and Norvig, *Artificial intelligence: A modern approach (second edition)*, Prentice Hall, 2003.

[37] Alexei V. Samsonovich, *Why BICA is necessary for AGI.*, Artificial General Intelligence, Proceedings of the Second Conference on Artificial Intelligence (Arlington, Virginia, USA) (M. Hutter B Goertzel, P. Hitzler, ed.), Advances in Intelligent Systems Research, vol. 8, AGI 2009, Atlantis Press, March 2009, pp. 214–215.

[38] Daniel L Schacter, *Implicit memory: History and current status.*, Journal Of Experimental Psychology. Learning Memory And Cognition **13** (1987), no. 3, 501–518.

[39] Alan M. Turing, *Computing machinery and intelligence*, Mind **LIX** (1950), 433–460.

[40] Jorn Veenstra, Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and akub Zavrel, *Memory-based word sense disambiguation*, Computers and the Humanities **34** (2000), 171–177, 10.1023/A:1002459020102.

[41] David Yarowsky, *Hierarchical decision lists for word sense disambiguation*, Computers and the Humanities **34** (2000), 179–186, 10.1023/A:1002674829964.

Vita

Justin Walter Deschenes

Education

Bachelor of Computer Science, *First Class Honours*, University of New Brunswick, 2009.

Publications

Bochem, A.; Deschenes, J.; Williams, J.; Kent, K.B.; Losier, Y.; , "FPGA design for monitoring CANbus traffic in a prosthetic limb sensor network," *Rapid System Prototyping (RSP)*, 2011 22nd IEEE International Symposium on , vol., no., pp.30-36, 24-27 May 2011

Last updated: September 18, 2012