The incorporation of human memory models into artificial intelligence techniques

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:	name1, degree, department/field, Chair
	name2, degree, department
External Examiner:	name, degree, department/field, institution

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

Month, Year of submission to Graduate School

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[**?**], the difficulty of the problem has prevented 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 (eg. 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 is required as the field matures. Although very little is known

about human sentience or our basic problem solving abilities, humans 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 us. 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, I looked into natural language processing became a good candidate and 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. 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 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 worthy topic for a thesis. 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, 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 larger, statistical methods perform poorly. A number of techniques have been developed to combat this problem; when these techniques are combined, small incremental gains in accuracy are achieved [**?**]. Modelling human memory may provide insight into improving accuracy when working with both large data sets and loosely coupled data areas. Humans using general intelli-

gence, past memories and other conscious and non-conscious mental abilities are able to instinctually discover minute patterns in large sets of data, quickly [**?**]. 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 problem), and early artificial intelligence work done in biological systems. It will also give an overview of some of the 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 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

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[28]. Natural language processing(NLP) is theorized to be AI-Complete [18], 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 [17], the application to restricted worlds and languages[25] 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 re-

lying 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 [19]. 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 [19]. 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; which 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

Although the high level overview helps to add understanding to the problem, the insight into the difficulty of natural language processing required many of the sub-problems to be researched first.

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 [5]. The problem was encountered initially in automated machine translation; in fact, it was one of the first "hard" road blocks [22]. Initial approaches considered using computers to automate the understanding of languages; however, at the time, there was an insufficient collection of data [22]. 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 contests outlined a number of challenges over the years; these challenges included different dictionaries and different goals as related to word sense disambiguation. 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 [11, 16, 29], instance based learning [21], decision lists [30], ensemble methods [13], kernelization [14], as well as various knowledge hybrid methods [23].

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 [24, 2]. The investigation into biologically inspired cognitive architectures opens up a number of areas [26]. 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:

2.1 Baddeley and Hitch

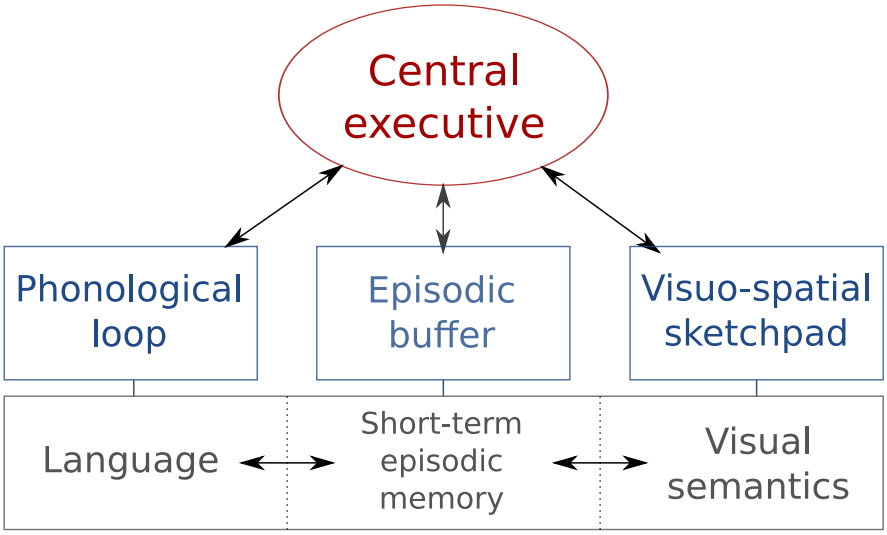

Figure 2.1: Baddeley and Hitch Model

In Baddeley and Hitch's model [3, 4], 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 unnecessary 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 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 the 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. The working 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"[4]. 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 Cowan

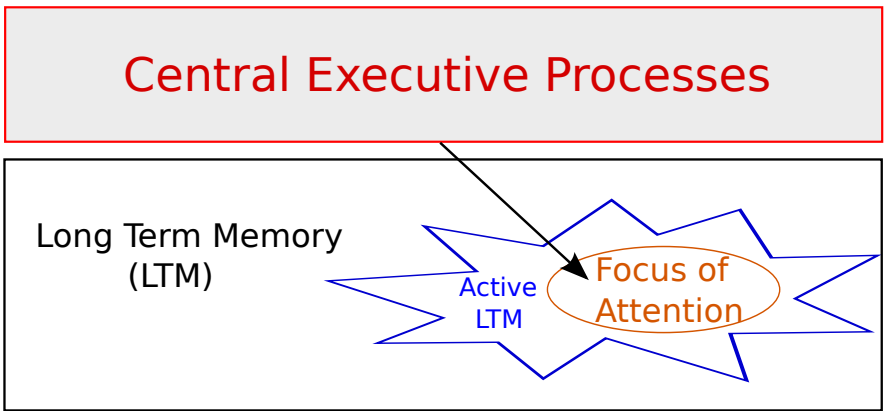

Figure 2.2: Cowan Model

In Cowan's theory [9, 10], 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 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 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. 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.

2.3 Ericsson and Kintsch

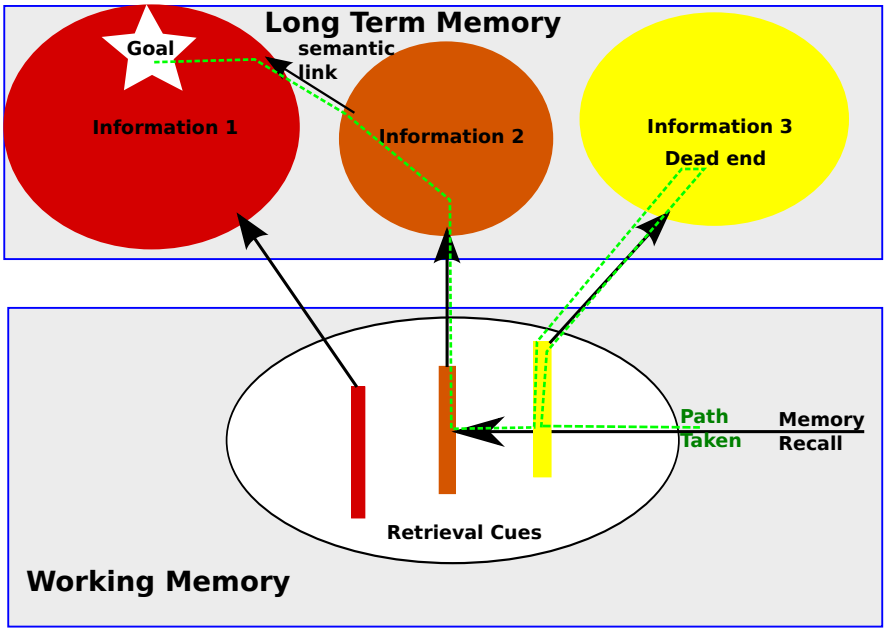

Figure 2.3: Ericsson and Kintsch Model

In this theory, Ericsson and Kintsch [12] 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 patterns and schemas, 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" of the long-term memory connected to it; this allows great memory performance by allowing rapid and flexible long-term memory access [12]. 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.

2.4 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 [15]: "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 [20], to, it is prevalent and strongly influences nearly all higher mental processes [7]. The effects of the unconscious has in its majority created the complex and intelligent design necessary for living things through adaption and natural selection [6]. 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.[8] 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 process and structure are not completely related to explicit memory [27].

While little work has been done in applying these models to any aspects of computer science or artificial intelligence, it seems like a natural fit. 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 is and the importance of it 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 helps 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 direction 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 architecting required to make the systems work together. This chapter attempts 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 3.

In the general case, psychological and biological models related to the human brain, and its learning capability layout 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 of the mind's systems are 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' 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, accuracy 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.

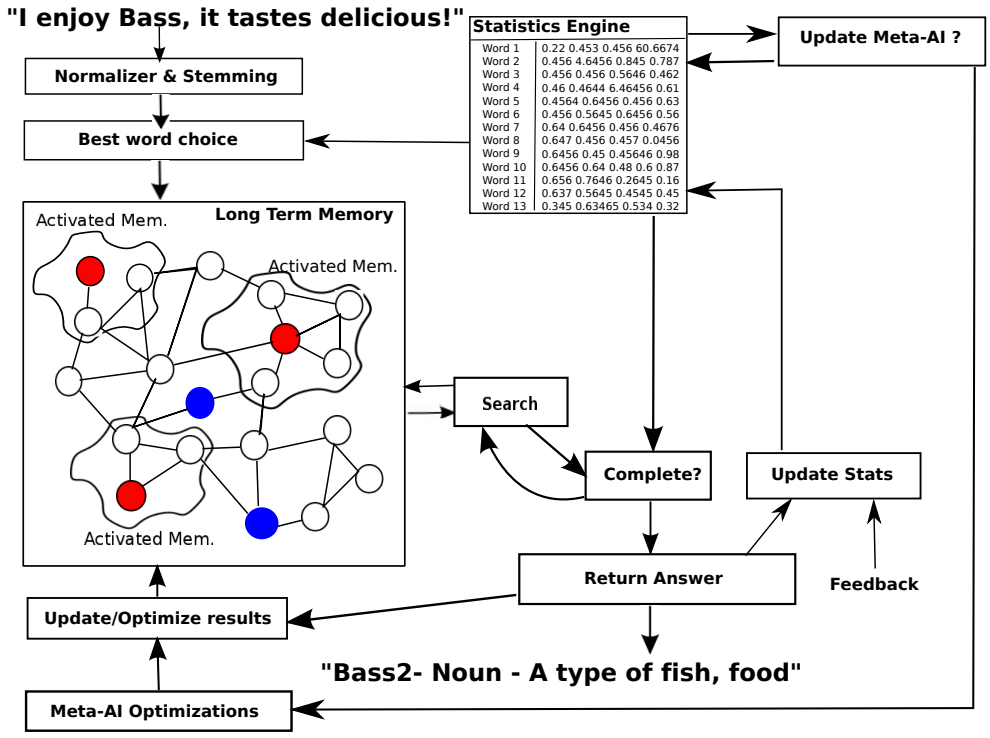

Figure 3.1: Proposed Design

3.1 High-level solution

Figure **??** gives a high level overview of the learning process. It begins by receiving a sentence and a target word to disambiguate, it then preprocesses the sentence by removing stopwords then normalizes and stems the remaining words. After this the sentence is run through a "best word choice" algorithm. This algorithm attempts to determine which words are the most "important" and how many of the words should be used to determine the sense. After the words are chosen they are then added to short term memory and loaded from working memory. From the working memory graph, the nodes are passed to a search function that simulates a cognitive thought process (linking memories from one to another) until a result is found. At the same time, a "gut check" or fast statistical method is used on the selected words to determine what is the likely disambiguation. These methods return their best guess and a confidence score which is passed into the meta-learning algorithm which uses the performance of both algorithms on the current dataset to modify the final disambiguation/confidence value. After the answer is passed on, the statistics collected for the current problem instance, through the use of the short term memory object is folded into the working memory and statistical engine. Finally, a check is done to determine if a meta-AI update is required. If it is required, the system evaluates the accuracy since the last update, and determines whether it should increase the current selected value, return it to a previous state, or randomly select a different meta value.

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 it differs from the ideal solution.

The current model consists of three functional components, a higher level container object and a "short-term" memory data structure. 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

The memory object could be considered the interface to learning, it contains methods which provide the following functionality:

1. a way to load and save the state of the mind. This way the learning environment, specifically the meta-learning information, but the other two components as well, can be stored for further refinement against additional datasets. 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 the mind, and attempt to optimize the learning of the data.

3. a answer method which queries the functional components to find the best answer given the initial information.

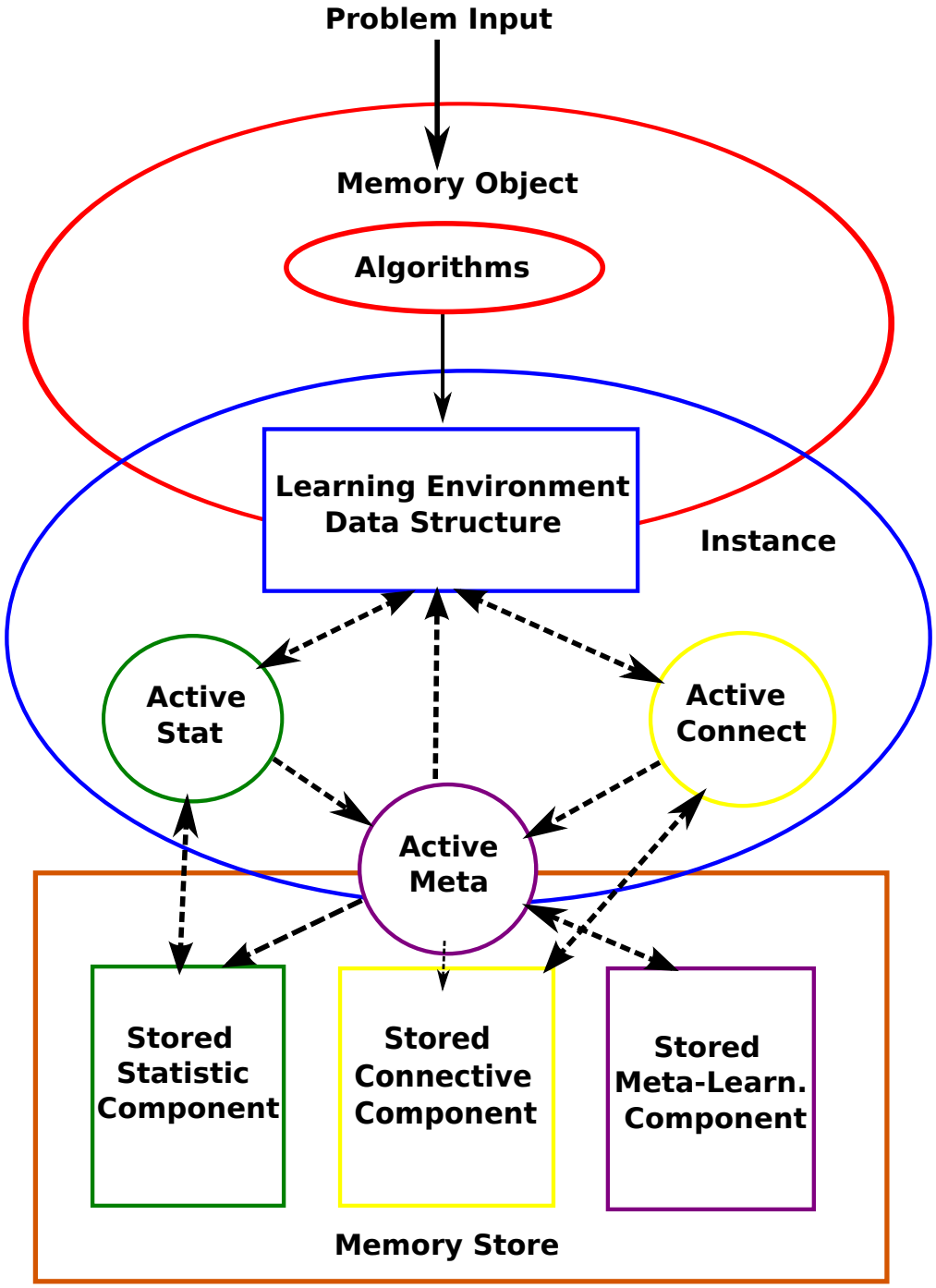

Figure 3.2: Component Overview

While nothing overly complex is going on, 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 subconscious 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 inline with the Baddeley's Central Executive.

3.2.2 Statistical component

The statistical component is an attempt to mimic the subconscious mind, a quick pattern matcher which is the first component of memory to have a possible solution to a problem. Tests have shown [**?**] the subconscious 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. While solutions come in the form of an often indescribable uncertainty [**?**], they are depending on your expertise in the area, often correct.
The statistical component is responsible for the following functionality:

1. a method to capture and train the 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, in which given a list of words and a target uses frequency and accuracy statistics of each word against each of the disambiguations of the target word

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 currently based on the percentage of times a word has been seen with the targeted word.

4. a method which determines best guess, or given nothing but the target word, what is the best disambiguation

Currently simplistic, it 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 subconscious mind would collect and index every piece of statistical information it could find, and using all non-crucial resources continually refine and apply new information to existing problems. This would allow the mind to give it enough information for it to make a "gut check".

3.2.3 Connective component

The connective component models long-term memory, a logically organized data structure which 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 decay, increasing the success rate of determining the deductive path.
The connective component is responsible for the following functionality:

1. a method to capture/learn the words. This method simply loads the problem instance's words as a complete network into the memory graph, augmenting existing nodes/edges as is required.

2. a voting mechanism, in this method each of the chosen words gets to vote on which disambiguation it believes is correct. The voting currently occurs through first target word found while traveling the memory structure using Djikstra's algorithm.

In addition to the above functionality, the data structure itself allows for feedback from the working memory object. This feedback currently allows nodes that were successfully/unsuccessfully used to find the solution to increase/decrease the edge weight between the selected node and target node. This is designed to effectively model the self imposed reward/penalty (sureness/doubt) which comes with properly or improperly using information in forming a solution.

The expressiveness of a graph in modelling long term memory does allow for a natural correlation. While the form 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 inline with the models:

1. dimensionality - currently all items are mapped to one dimension, the node's edge weights are valid for the entirety of the problem set. It is my understanding that the edge weights should be biased on meta information used in defining the problem. This information would segment certain components of the edge weight, allowing the graph to be modified based on the context of the problem. Currently this was set aside as the graph is already slow enough, and the meta-learning component is not able to learn/tag individual problems as related to other problems (note: although it would be a good place to implement an unsupervised learning algorithm (unsupervised decision trees comes to my mind first)).

2. Subconscious access - as mentioned above, in a parallel environment similar to the brain's, ideally the subconscious working away would effect 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 problems solving to the current model. It is responsible for various meta-values which are added into the algorithms of the other components. Currently there is only a few, however this reasonably outlines a simplistic model of how the building block functions, or even executive functions could be initially modelled.
Currently tracked values include:

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

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

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

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

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 simply randomly selects a value to change, changes it 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 the their voting confidences based on past performance. Ultimately giving the meta-learning component 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 amount of resources (massive parallelism again) required, the necessary information to track would grow, and more importantly, the amount of training examples required to generate good results would be a very large number. Initially genetic algorithms just to get off the ground, but tackling the executive function problems, and then the specific component algorithm creation requires something outside of my current thinking.

3.2.5 Learning Environment Data Structure

This object simply tracks required information to solve a problem. It is in a sense a simplistic short term memory object. It is required to be passed into all of the components as each component requires tracking information. It is then used to modify the Long term memory structures.

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 to better understand the complexity of the model. This section analyses the components against the models, shows their short comings 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. The functions that made up the original executive got moved into the memory workflow. Methods which coordinated the systems 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 the need for dynamic execution was no longer required.

The looping components of the model was reduced from the three or four components outlined into one main looping structure. This looping structure consisted of the train, optimize and guess loop. This 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, is also quite slow for the non-parallelized program (and even then work on handling simultaneous read and writes would be required). Finally the guess loop uses the metrics gathered and the longterm 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 them 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 at the same time 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, 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

are available to the mind as a whole, thus compounding the learning ability and increasing the likelihood of making longterm and unconscious connections. With Baddeley and Hitch specifically stating that these components work in parallel with one another, with any information passing, or issues handled by the central executive, the learning and understanding potential is much more powerful then 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 components 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, and 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 which 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 which 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, which when the problem is completed, the active nodes are returned to the long-term store, with new problem solving statistics.

One of limitations of the implementation in regards to Cowan's model, is the statistical collection and 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 statical 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. Where as 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 attempts to load the memory objects (and therefore cues) which 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 components 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 components 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 what the subject matter contained is. 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 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 important as every other item. It does use the 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 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 subconscious 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 conscious mind does, the experiments talked about in (LOCATION OF BLINK EXPERIMENTS) show that some form of statistical analysis is performed and it is of great use in decision making. The idea of the subconscious as a statistical solution engine guided the design. The studies on the use of unconscious mind emphasized the difference between declarative knowledge which allows one to articulate reasons for choices and procedural knowledge which allows repeated complex actions to be understood on a unconscious level. Where declarative memory takes longer to recall, the mental processes which act on it allow more useful relative information to be made available. Procedural knowledge on the other hand can, over time move from a subconscious 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 which 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. Although unlike true procedural knowledge, the implementations version has a much more conscious process. Currently, the model relies on a predefined collection of statical information. This information is then used in a number of predetermined algorithms, some like the hypothesis testing algorithm allows for flexible and adaptable use of

information 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 redefining multiple hypothesis, 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.
- 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 the algorithms that make it up. 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. Where as the last chapter brought up the design decisions and the reasons behind them, this section deals with how the system implemented the ideas. This section also discusses the outcomes of the various implementation choices and attempts to further justify the design decisions.

Before jumping into the specifics of the test implementation, a quick reminder of the problem domain is in order. Given a sentence, 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 was taken from the Natural Language Tool Kit (NLTK), an open source python project for use in linguistics and natural language processing research and development. 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 dataset. The dataset is made up of manually tagged sentences made publicly available through the Senseval/Semeval workshops, a workshop put on by the Association for Computational Linguistics which promotes 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 dataset consists of 15225 sentences containing an average of 27 components. The sentences were organized in a strict XML format, with each record 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”. 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 the targeted word is found in. 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', 'PR`

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

In addition to the dataset, 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 was setup using five-fold cross validation in which four partitions of the dataset were used for training, and the fifth was used for testing. Which was then repeated five times, allowing each of the data sets to be the test data, and having every combination of use for the training data. This method

differed from the method used at the Senseval-2 competition, but allows for an acceptable comparison. The Senseval-2 competition gave the researchers a full training set, but held back on releasing to them the data they would be testing.

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. The statistical component collected predefined information which was to be used later for decision making. This first phase ran over the entire training data.

Once this initial processing was done, a second pass of the training material was required. This pass emphasized the learning process. In this phase the graph component setup in the previous phase was further adapted and trained to the dataset. The statistical component was tested for accuracy and the meta-learning component began its initial setup and data collection on information from the accuracy of the two other components. This in-

cluded having the meta-learning component query the statistical component to learn which words in the sentences were the best to disambiguate on. This process attempted to eliminate noisy, or unimportant words and select only those words with the best chance to properly disambiguate the target word. Selecting the best words was done by using an algorithm which made use of the following three factors:

- Completion rate: this was determined by how likely the word was to complete successfully. Which could also be considered the relationship between the word and the successful selection of some word sense in both the statistical and connective algorithms.
- Accuracy: the probability the word would be accurately disambiguated.
- Rarity: the occurrence of a word

Finally, once the best words were known, the number of words to choose per sentence was then determined by the meta-learning system.

Another important aspect of the training phase is the reporting method. This is done through the use of a Learning Environment Data Structure (LEDS). This data structure carries the information used to solve the problem, collects the statistics and computes the meta-data. This data structure is passed into both the statistical and connective guessing process, it collects their choices and other data required to be updated after the learning instance is completed. It is then passed into the meta-learning phase, in this phase it

is used to help compute the final answer. Finally, after the algorithm has made its guess, it captures the correct disambiguation, and is used to update the long-term memory data structure with the changes noted in the learning instance phase.

4.3.1 graph-based adaptation

The method in which graph-based adaptation took place was through the use of collected meta-data, and instance based feed-back. With the initial graph setup, each word in each sentence were interconnected. Over the training data this formed a very dense graph, with the value for each edge allowing for the most likely location for a given word sense to be found. The idea behind the algorithm was each word selected as important in a sentence would vote on which disambiguation was most likely, the word sense with the greatest number of votes would be determined as correct. This technique not only determined the "best" possibility, but gave an approximate confidence value in the chosen word. The search performed went through each of the selected starting words, and would look through its neighbours for the given word senses in decreasing edge weight order. If an occurrence was found, it was selected as its vote, otherwise it would pick the highest rated edge and perform the same search on its neighbours. If after a meta-defined number of neighbours were checked, and no disambiguations were found, the selected word would not vote. If in returning from the search there was a tie, the vote would be decided arbitrarily. Throughout this search the path each

word took and the disambiguation it voted for was tracked and kept to be used to adapt the graph data structure.

The graph adaptation algorithm was performed after the instance problem was completed and the answer was known. During this phase a few outcomes were possible:

- the graph component correctly determined the word and each word completed its search. In this case each edge along the path each selected word took had its weight increased by a meta-determined amount.

- the graph component incorrectly determined the word and each word completed its search. In this case each edge along the path that led to the wrong outcome would have its edge weight value decreased by a meta-determined amount. This amount was often significantly larger then the increase value, in an attempt to offset initial training imbalances. The remaining words, if any, would not have their values adjusted.

- the graph component correctly guessed the word, but the majority did not finish their search. In the case where confidence is low, the graph component would not submit a vote, but would keep its best guessed value to help determine feedback. In this case each edge that completed with the correct value had the value of the edge weights along its path increased (usually by quite a bit more then a normal correct case). and the edges that did not finish would have the edge weights along their

edge decreased by a significant meta-determined amount.

- the graph component guessed incorrectly and the majority did not finish their search. In this case each of the values of the edge weights along all of the paths would be decreased by a meta-determined amount, which in most cases was usually the largest reduction.

After completing the second run of the training dataset 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 beneficially, 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. The best method for this, and one that seemed directly related to the actual process was Bayesian statistics. After each learning instance the statistical information contained there in 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, enumerating each sentence and each word in the sentence to its proper disambiguation, and the disambiguation to the various words seen.

The second phase, the optimization component, the statistical component attempted to select the correct disambiguation based on prior knowledge, and would again add the information to its statistical collection. The selection algorithm initially used a wider range of variables to determine which word sense was most likely, this however proved to be a poor choice, as the standard selection algorithm (most frequently selected) worked much better. This led to the second iteration, the use of the 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 returned the disambiguation which has the highest occurrences. As mentioned above, the statistical component is also used to select which words are the best words to use in a sentence to determine the disambiguation. This information is collected in both phases, but only used in the second phase.

4.3.3 meta-data collection and adaptation

During the optimization component of the learning phase, the meta-data collection and adaptation is completed. This phase consists of two important parts:

- answer biasing- This phase adds a bias to using the answer provided by a component in accordance with how accurate the component is. This uses information collected through the meta-data analysis of all

previously answered learning instances to give weights to each of the answering components equivalent to their accuracy.

- adaptive processing- This component monitors and tests the pre-defined meta-values. At each phase, a meta-value is randomly selected from the list of all meta-values, it is incremented based on a pre-defined "step" value and then is run for a number of instances. If after the test run the changed value being tested performs better than average, it uses this new value as a baseline, if it performs exceptionally it attempts to retest the value with an even larger value. If the changed value performs worse it reverts the value, if it performs exceptionally worse, it attempts to retest the value with a negative "step", a smaller value.

The meta-variables these tests were performed on consists of two groups. The first group being the "desirably controlled quality" group. This group consists of values not directly related to accuracy, but likely have an effect on the results. The main reason these were added was to have the system optimize itself throughout the learning phase and into the more efficient 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 slower, while maintaining or increasing accuracy; Or when it should use less words and run faster without effecting accuracy.

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

- graph depth - the maximum depth 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 "inflation limiting" group. This group consists of values that can directly affect accuracy, and were added to the meta-variables specifically to counter inflation of edge weight values over time. The hopes were this would combat dataset biases.

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/cognitive aspect of memory and learning, with a great chance of increasing 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 in the "zone". 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. The relative success of the work will be outlined and finally a section containing guidelines on what research and future work can be undertaken that could lead to better results, or a more accurate model of human memory.

5.1 Quantitative results

With an overall average of 59.65% this test fell just within a standard deviation of the baseline result of 56.94%. However the majority of the candidate systems from the Senseval 2 competition performed within the standard de-

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

Table 5.1: Results Table

viation or worse 5.2. 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 competitions baseline test, validating both the datasets, and the testing approach used.

The information in the regression analysis of training accuracy **??** holds the results of all instances over the entire 5-fold cross validation process, each fold consists of approximately 12140 training instances which were recorded and 2428 instances which were not included as they were used as test data, and included in the test results. The resulting graph shows two trends that require discussion. The first trend is shown with the red line, this shows the steadily improvement over the run of all tests. The 2.5% gain on the average accuracy over the 5-folds can be contributed to the meta-learning component. The meta-learning component adjusted the meta-data that controlled how to train the three learning components. These values carried over to each of the proceeding training instances. The other important trend is

English all words - fine-grained scoring			
precision	**recall**	**attempted**	**system**
0.69	0.69	100%	SMUaw
0.636	0.636	100%	CNTS-Antwerp
0.618	0.618	100%	Sinequa-LIA - HMM
0.597	**0.597**	**100%**	**Current technique**
0.575	0.569	98.91%	UNED - AW-U2
0.573	**0.573%**	**100%**	**self-created baseline implementation**
0.569	0.569	100%	Baseline
0.556	0.55	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)
0.5	0.449	89.729%	UCLA - gchao
0.36	0.36	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
0.44	0.2	45.37%	Sheffield
0.566	0.169	29.883%	Sussex - sel-ospd
0.545	0.169	31.055%	Sussex - sel-ospd-ana
0.598	0.14	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, current technique and self-baseline implementation added

found in looking at the general improvement of values over one test fold, to make it more clear the regression analysis of one fold has been completed **??**. The important thing to note over the training data set is that the statistical component was pre-optimized during the training 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. While the meta-learning changes are cumulative over the whole graph, the nearly 5% increase in accuracy over the course of a single folds graph can be credited to the training of the connective component. Furthermore, as the accuracy increase 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. Generally then, we can relate the increase in accuracy over a single test fold to the improvement of the connective component, and the minor tweaks to how the connective graph and the meta-learning component adapts to the changes the meta-learning component creates. In both trends we can say statistically that the accuracy and number of instances are positively correlated.

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
r^2	**adjusted** r^2	**F-value**	**MSE**	**AIC**
0.0882	0.072	5.42 (p = 0.0235)	$4.57 * 10^{-4}$	-278

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

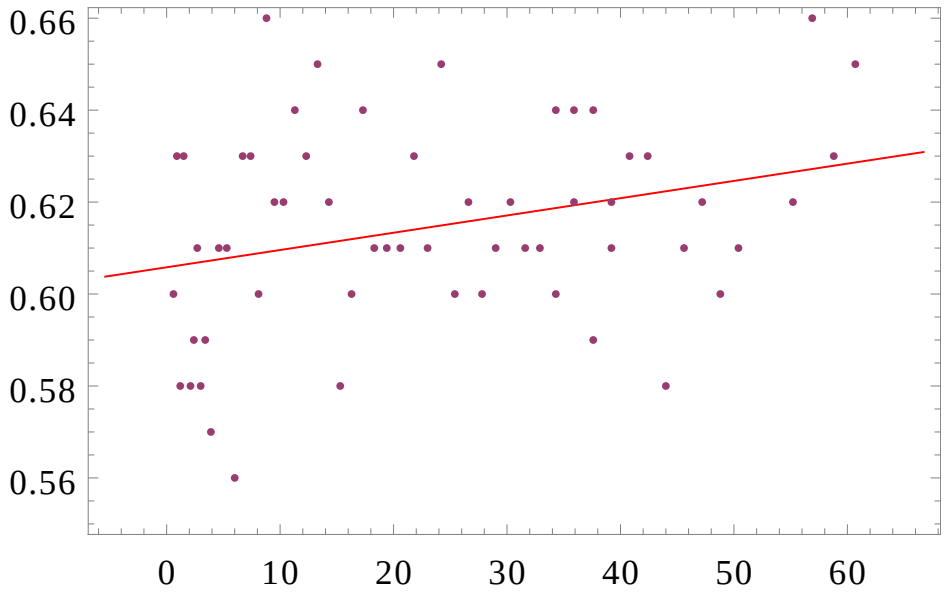

Figure 5.1: regression analysis of training accuracy (x Instances x1000, y Accuracy)

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
r^2	**adjusted** r^2	**F-value**	**MSE**	**AIC**
0.24	0.2	6 (p = 0.0242)	$5.33 * 10^{-4}$	-95

Table 5.4: statistical values of regression analysis over one fold

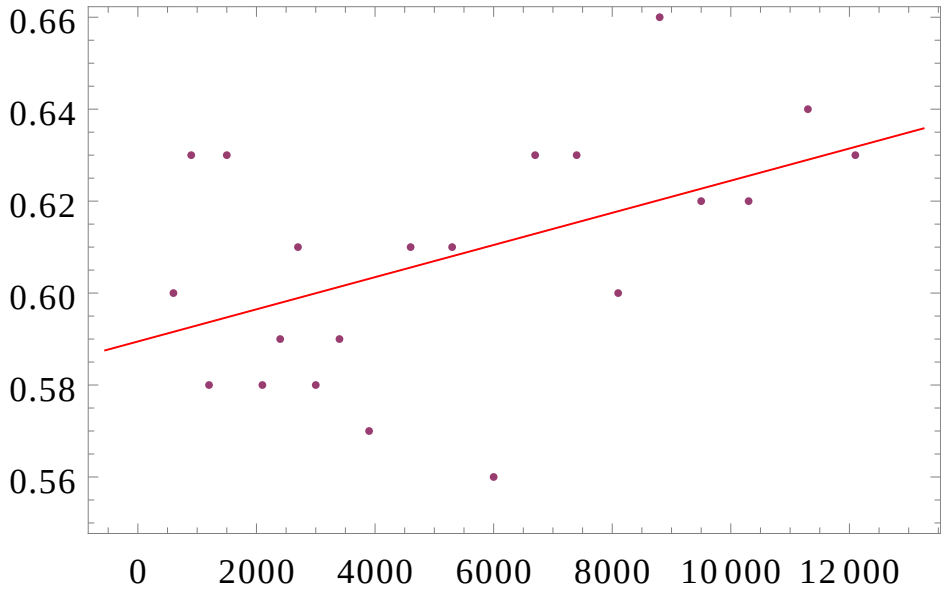

Figure 5.2: regression analysis of training accuracy over one fold (x Instances, y Accuracy)

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 5.5 we see the total results for all of the training of the cross-validation runs. In total 60700 training instances were run, of which 37587 were successful. The statistical component was correct 59.58% of the time, which made up 96.22% of the correct results. The connective component was correct 37.26% of the time, and made up 62.5% of the correct results. Although 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 instances	**correct**	**percentage**	**connective correct**	**stats correct**
60700	37587	61.92%	22618	36166

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 op-

timally and chose the correct value from the components every time a correct value was presented. This table and the information presented earlier clearly show three 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 then 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.

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 then 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, 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, where as the testing results have both of these locked in. Quicker feedback results in better accuracy, and was done to simulate human testing feedback. Where as to follow the rules of the Senseval 2 test, this mechanism was not used during the testing data. This trait appears to positively effect the connective component's accuracy, but has very little effect on the meta-learning component.

training results					
acc.	**statistic acc.**	**excl. stat.**	**connective acc.**	**excl. acc.**	**optimal 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.	**statistic acc.**	**excl. stat.**	**connective acc.**	**excl. acc.**	**optimal 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%
training 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 biasness

5.2 Qualitative results

The results are competitive with other systems from the Senseval 2 test, it performs respectfully if a little low given the 10 years that have passed. 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 future work and understanding required to improve its likeness to human memory and overall accuracy.

The outcome and results show a successful interpretation of the human memory models discussed in earlier chapters. In table **??** 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 systems 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 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 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. The increased accuracy in using differing views on the data provided greater overall accuracy. Although simulating additional senses from the data provided by the Senseval 2 datasets 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 that 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 "gut check" to 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 for adapting working memory into long-term work together. 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 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 weighing 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 datasets, 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.

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

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

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

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

[7] 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.

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

[9] Bruce G. Buchanan, *Timeline: A brief history of artificial intelligence*, May 2012.

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

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

[12] 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.

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

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

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

[16] 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.

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

[18] 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.

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

[20] 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.

[21] 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.

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

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

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

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

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

[27] 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.

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

[29] 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.

[30] D. L. Schacter, *Implicit memory: History and current status.*

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

[32] 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.

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