WORD SENSE DISAMBIGUATION IN WEBPAGES

Developing a program capable to disambiguate words with a website text as context

Master thesis for Andreas Sekkingstad

Institute of Information- and media Science

University of Bergen

Spring 2016

ase032@student.uib.no

asekkingstad@gmail.com

Key Words: semantic web, semantikk, nlp, WSD , Wordnet

ABSTRACT

This master thesis investigated automatic methods of Word Sense Disambiguation (WSD) in HTML pages. The hypothesis was that HTML documents provide various disambiguation cues which are not normally present in general text, and which can enhance the quality of WSD. We tested several existing natural language processing toolkits which provide general WSD services, and compared these to our novel algorithms which were designed to take advantage of the HTML cues. The findings showed that our new algorithms outperformed state of the art general WSD implementations. In addition, our algorithm could provide a ranked list of potential disambiguations, which is useful in an example use case where users "tag" key words in a web page with the help of the disambiguating algorithm

ACKNOWLEDGEMENTS

Firstly, I want to thank my girlfriend and mother to my child for helping me through tough times when working long nights on the master thesis.

Secondly I want to thank all of my testers and questionnaire do-ers. You have greatly helped with enduring the lengthy and trying task I asked you to do.

Lastly but not least, I want to thank Csaba for supervising and pushing me in the right direction when work progression halted.

1 INNHOLD

Appendices

Figures

Tables

Equations

Abbreviation List

WSD- Word Sense Disambiguation

POS- Part of Speech

NLP Natural Language processing

IR- Information Retrieval

MT- Machine Translation

WN- WordNet

NLTK- Natural Language ToolKit

AI- Artificial Intelligence

NER- Named Entity Recognition

NGD- Normalised Google Distance

HTML – HyperText Markup Language

2 INTRODUCTION

Word Sense Disambiguation is an NLP task of assigning the correct sense to a "Target" word based on the context in which it is located. In other words, extract the words meaning within the context, use contextual clues to assign a sense to the target word. Traditional Knowledge based WSD methods uses the surrounding context words senses and compares them to the target words senses. Comparing the senses is usually done with sense relatedness approaches (Lesk 1986; Wu and Palmer 1994b; Lin 1998).

According to WordNet's definition of a "word sense" it is the accepted meaning of a word, however in the Wikipedia entry of "word sense" it is: one meaning of a word. There are of course several other sources of definitions, the main idea is that a word has different definitions in different contexts. Which is the thesis' main question, the one of ambiguity in words. One word can have several different senses, having more than one sense is part of the definition of having the property of polysemy. Differentiating between the senses is the problem of ambiguity. However, the problem of locating the correct sense of a word in a context is not only a problem situated at word level, but at sentence level. A sentence can have lexical ambiguity and structural ambiguity. Both with polysemy and without, meaning that we can know every sense of each word in a sentence and still the sentence can be ambiguous. On the other hand, the most usual problem in word sense disambiguation is sentences which contains words with polysemy, and the fact that the sentence makes perfect sense with different senses. It entirely depends on the context. Previous work in the field of WSD consists of both task dependent and independent work. For example, that in a task dependent, domain-specific with a knowledge based WSD systems performs better than generic supervised WSD as proven in Agirre, Lopez De Lacalle, and Soroa in 2009.

The research in WSD has been around for a very long time, as far back as the 40's. Warren Weaver (1949), was the first to present Word Sense Disambiguation. Weaver presents a solution to WSD when trying to figure out which meaning the word "fast" would have in a sentence. Knowing the sense of the word was impossible without context, and surrounding "fast" are N number of words, and to find fast's meaning one were to start with N=0 and expand N until enough sentence words could reveal fasts meaning. The date on this example makes Word Sense Disambiguation a very old, if not the oldest NLP problem, and is still researched to this date.

This thesis' effort to solve the general WSD problem, alas the task is not to find the exact sense that a word has in a context, but assisting users with the most likely of senses according

to the web context. And in this effort I have developed novel WSD algorithms that attempts to assist in work such as Semantic tagging by eliminating the cases where homonymy and polysemy cause problems.

The algorithms developed in this thesis will be able to assist in areas such as online semantic tagging. Semantic tagging was an integral part of "Web2.0", and was called social tagging. Used in content rich sites such as FLICKR and delicious. The act of "Tagging" is: assignment of uncontrolled textual metadata to resources. Examples of such tags could be single word or two word collocations. Very often the tags were abbreviations and various idiosyncratic concatenations (e.g. "toread"). But with uncontrolled strings like so, it brought upon problems like ambiguity and synonymy. Meaning that one-word form can encode many concepts and the same concept can be encoded in many forms. Work in the field, uses semantic tags instead of simple unconstrained strings. The semantics of the tags can be comprised of Wikipedia entries from DBPedia, and electronic lexicons such as WordNet. LexiTags is an example of such an application, LexiTags is a bookmarking application which uses semantic tags instead of unconstrained strings. Usually, people have manually tagged their own websites with available metadata, like mentioned above, which is a place where WSD in webpages can come in handy. More or less automating the disambiguating process, where one would previously have decoded the ambiguity manually.

2.1 Thesis Overview

The thesis is divided into eleven different chapters, the first chapters presents the thesis introduction with the following research question, and the thesis' goal for the completion in this thesis.

The second part will encompass the theory in NLP and WSD explaining what the field of NLP is about and in detail WSD. Some problems within the field is covered along with previous research. This thesis emphasises the use of WordNet in the development as well, so a minor section presenting WordNet and its usage is also in this section.

The second part of the thesis will explain the development stages during the thesis. The algorithms implemented in the project are both originally made in this thesis and pre-existing ones made by others. The stages ranges from pre-existing Word Sense Disambiguation software to what I have built for this project. Testing and researching the algorithms built is also within this part of the project. Every Result from the testing phase is presented in the section named "Results".

The final part will be part Discussion, Conclusion, answering the research questions, and tying together the hypotheses posed during the thesis, followed by, further work, tools, references, and appendices.

3 RESEARCH QUESTIONS

Q1: Can existing WSD algorithms accurately predict the correct sense of a target word in a webpage?

To answer this question, I have researched and implemented existing algorithms. Testing the disambiguation capabilities when faced with context and target words collected from different web pages. Implementing these algorithms and measuring them against human WSD tasks, measures their capabilities. Testing data comes strictly from different webpages.

Q2. Can we construct more accurate algorithms by considering the standard HTML elements like titles as a contextual element?

This required me to develop WSD algorithms that weighs HTML elements differently. The HTML element in question besides the word that a user has tagged and the surrounding text, is the title of the web page, or the HTML heading.

Q3. Can we use the disambiguating algorithms to assist users in tasks like semantic tagging, or other forms of manual markup?

Building algorithms that can handle the ambiguity that comes with textual content in HTML pages is the main issue in this question. Algorithms that resolve this ambiguity and has the ability of returning a short list of most probable senses based has been built to be of assistance when reviewing the correct meanings of words when marking up online content.

4 BACKGROUND

This chapter will reveal the general research and techniques done in the fields involved in this project, namely NLP and WSD. Techniques and algorithms used in the project is also discussed, this is important for the hands-on development of this project. Mapping out ideas, researching possible solutions, and pre-existing algorithms available to use is key in the start of the thesis.

4.1 NLP Natural Language Processing

NLP is a field of many divided tasks. Examples of such tasks are: Automatic Summarization, MT, Named Entity Recognition(NER), Natural language understanding, POS tagging etc. The field has branches in computer science, linguistics and artificial intelligence. The main concern of the field is the divide between human and computer language. Collobert and Weston (2008) state its aim to be : (...)"to convert human language into a formal representation that is easy for computer to manipulate."(p.1).

My thesis is based on theories and methods from NLP such as word similarity measures, WSD techniques, and technologies to help me answer the research question and reach the goal. Researching natural language processing software to aid in my development is necessary to create an end result capable of efficiently distinguish word senses from another. The mentioned relevant tasks in NLP are presented in the subsections below.

4.1.1 PART-OF-SPEECH TAGGING

In sentences there are different types of words and in different parts of a sentence they have different functions. Some examples of these different parts are nouns, verbs, adverbs, and adjectives (a more specific tag list is available for review in the appendices). POS tagging is the act of marking the individual words in a sentence to their corresponding word-category. The tagging has been done by hand all the way to rule-based algorithms, an example of such a tagger is E.Brill's tagger(1992). Developing the program, it became clear that some words could be used in most of the available sentence elements, and one example is the word "light" which has more than seventy different senses. And excluding senses that is in a different part of speech form decreases the work that a disambiguator has to do. This can be extended to the given context that a disambiguator needs. If the target word is a noun and word number four in the context is a verb. Given the algorithm, one can remove the whole noun, adverb, and

adjective part of that particular word and measure the relatedness between the remaining senses. Even state of the art Part-Of-Speech taggers (ACL 2013) that have an accuracy close to 100 %, can be wrong as well. And especially wrong in the cases of sentences(Manning 2011), this article explains that in the use of Part of Speech taggers in sentences really have an accuracy of 55-57 %. Which could prove disastrous for POS tagger usage in conjunction with WSD methods, especially if sentences have spelling errors or if there are informal speech such as slang. Researching available software to use will also become a part of the thesis, the question of whether I should build my own or use a state of the art POS tagger is connected to the amount of time available in the project and the focus of the thesis. The focus is on WSD and not on building a POS tagger, and the time available should be used to develop and test WSD algorithms.

4.1.2 WSD – WORD SENSE DISAMBIGUATION

WSD is an open problem in Natural language processing. The field is dedicated to identifying the semantic properties of a word within a sentence given the different contexts. Even though it being an open problem in the NLP community, one asks the question: "What is it used for?".

WSD has been around since the 1940s where researchers have created more and more complex methods over the years. More accurate methods for machine learning and manual methods have been constructed in the 2000s, making WSD still a topic for AI and linguistic researchers. Approaches and methods include shallow approaches and deep approaches. Dictionary and knowledge based methods, primarily use predefined knowledge like thesauri etc. Semi-supervised, supervised, and unsupervised are methods to be researched for this project to be successful. So far the research has found that a WSD method called the shallow approach with methods such as collocations and co-occurrences. Ted Pedersen (2000) that: (...)" shallow lexical features such as co-occurrences and collocations prove to be stronger contributors to accuracy than do deeper, linguistically motivated features such as part-of-speech and verb-object relationships." (p.6). This can act as a starting point in developing an algorithm with capabilities needed for the thesis, hopefully this will prove fruitful in the development. The prototype or software this master thesis produces will not go deeper than atomic understanding from context; meaning it will not go deeper than meanings of single words within sentence contexts. Literature containing information from earlier work within

WSD can give sufficient knowledge in developing an artefact with the capability of retrieving the correct sense in web contexts, a situation where Natural Language is common.

4.1.3 AMBIGUITY VS POLYSEMY

There are two kinds of ambiguity: Polysemy and homonymy, and as explained in the introduction there is a difference in ambiguity and polysemy. Ambiguity can be lexical or structural, lexical ambiguity happens on single word level. When a single word which sounds and is written the same but can be interpreted with different meanings, i.e. identical words with two or more different meanings. An example of this is "bright", as in an intelligent male or female, and that the sun shines bright today.

Structural ambiguity refers to the fact that sentences can include two or more interpretations from the same string of characters. A sentence with structural ambiguity lies not with a words lexical ambiguity, but in the way a sentence is built. This gives way for different interpretations. One example is as follows: "The woman saw the man with binoculars". Every word in the sentence is unambiguous, even if they have more than one meaning. Even though we know each words meanings, the sentence can still be interpreted as a woman that see a man through her binoculars or she saw a man carrying binoculars. So, every word in that context is specified, the words are not individually ambiguous based on the context. Still, it can be interpreted in two ways. Structural and lexical ambiguity both envelops the problem of several interpretations in identical strings of characters but on different levels.

Polysemy is when a word has two or more senses or meanings, but the fact that there are more senses is not enough to be polysemous. The senses have to be connected in some way as well as being clearly separable. There are a lot of polysemous words and an example of one is "earth", it can refer to the planet earth, soil, dirt, or ground. Though the meanings are different, they clearly have common features relating to each other.

Homonyms have different unrelated senses or meanings under the umbrella of a word(Dash 2001), where a polysemy's senses though different, they are related. An example of Homonyms is "stalk", which can refer to the act of stalking a person or prey, or the stem of some plants. Both the senses come from the same identical, same spelled word, however the senses are clearly very different. Also, if one looks up items in a lexicon, words with polysemy are listed as a single line with their meanings numbered below or beside. Homonyms are usually listed in separated lines in dictionaries, i.e. Listed as different words

with the same spelling. Homonyms differs in a lot of ways(Dash 2001), and some of the ones are listed above. The difference is subtle but important to note.

Ambiguity and polysemy are different, disambiguating words in ambiguous sentences involves finding the correct sense for the word according to the context. If the words in a given sentence had only one sense, then it would exclude the chance of polysemy, but not Homonymy. Would it still be ambiguous? With structural ambiguity it could still have the attribute of being a homonym (as explained above), and by definition, exclude lexical ambiguity in our particular example. Polysemous words can then be on the same level as lexical ambiguity, meaning that with several senses a word can mean any of the senses. Of course, there can also be polysemous words at sentence ambiguity, but it would seem like it would not need to include polysemy to be structural ambiguous.

4.2 Word Sense Disambiguation Area of Use

What is word sense disambiguation used for? What is the meaning of researching the field of Word Sense Disambiguation? In Agirre and Edmonds (2006), where the WSD field fits in is referred to as a means to reach other goals in computational linguistics and NLP. In other words, WSD should not be the end goal, but a means to increase performance of other fields or techniques, such as MT (Machine Translations) (Specia et al. 2005) . Machine translations is the field of translating written text or spoken, from one language to another. On that level, Word Sense Disambiguation can reduce the errors in translating from one context to another. Words can have different translations from different languages based on the context. One example can be the word “break” in the English language, in Norwegian it can mean “brudd”, or ”hvile”. As in: “broke the vase” and “take a five-minute break from work”.

From the same paper it is referenced from Vickrey et al. (2005), that a WSD module significantly increases performance in their statistical Machine Translation. However, there is also evidence that it does not increase performance, at least not a significant one. With the thought that WSD software should not be invented as a stand-alone generic all-use method, the thesis I am working on is on track. Having a task specific WSD invention, designed to solve disambiguation problems within a confined area.

There are other areas such as Text mining, parsing, information retrieval(IR), and lexical knowledge acquisition (Agirre and Edmonds 2006) that the WSD field can contribute to, presented below is WSDs use within Information Retrieval and Text Mining.

4.2.1 WSD AND IR

It is well understood that ambiguity is the base problem of WSD, and the simple understanding of WSDs tasks, achieving one hundred percent correctness in choosing the correct sense is yet to be a fact. And, mentioned above is the fact that WSD is task dependent and not a general problem to solve. One of the tasks is using WSD methods in Information Retrieval (IR), IR is retrieving relevant information from some large corpora of for example texts. Places that can have IR systems can be libraries with search systems that aims to find relevant texts based on your search. The retrieved list of texts in this case, is often a list of ranked results, where the ranking is most relevant to lesser relevant. Often in IR retrieval systems a query assigns the search results with a number to find which text is the most relevant.

WSDs role in IR is the fact that the search string can include ambiguous terms, following that, the ambiguity in the documents as well can hurt the precision of the retrieval(Zhong and Ng 2012).From the same paper it is also said that the query words can have related meanings with words outside the query. So by these hypotheses, developing IR specific Word Sense Disambiguation algorithms will help to rid the query and search documents of ambiguous results.

4.2.2 WSD AND TEXT MINING

Analysing text to obtain information is the general task of text mining, (Hearst 1999) mentions that text collections are “virtually untapped” due to its uncategorized and difficult encoding; that this is the reason why it is not extensively researched. Text mining attempts to discover patterns in text collections, patterns that can uncover information not visible to computers or people. Some situations where text mining can have value is for businesses that require news and live updates to thrive, such as stock investors. Online media, Facebook updates, Tweets, and news updates are all such examples. These are examples under the umbrella of unstructured data(Kanimozhi and Venkatesan 2015) . Natural text written in such medias are often ambiguous due to slang, dialects, age groups, and so on. Parsing text for Text Mining research will eventually encounter sentence ambiguity, and potentially parsing inaccurate information from sentences or sections within text collections. So empowering a Text Mining software could increase the accuracy of information extraction in both human and computer created texts. An example of such a software could be for law enforcement, the

need for alerting authorities when possible flags are raised in emails or social media updates. Flags such as "drugs" as in "illegal drugs" can be misinterpreted as "medical drugs". Word Sense Disambiguation can help distinguishing ambiguities in such examples. Though this example leans more in the direction of Information Retrieval(IR), it still holds in Text Mining. Word Sense Disambiguation has been proven to work in Biomedical Text Mining, techinques built to outperform others in terms of accuracy as well(Pesaranghader, Pesaranghader, and Mustapha 2014).

4.3 WordNet

WordNet is a lexical database which is considered to be a vital resource for computational linguistics(Fellbaum 2000). Its large lexical database consists of English words grouped into sets (Synonym sets). Nouns, verbs, adverbs, and adjectives are the type of words in these groups. The Synsets mentioned are interlinked in a conceptual-semantic and lexical relations. This means that words and senses are grouped together in way of their meanings, much like a thesaurus. The synsets are connected by a means of relations, examples of these relations are Hyponyms, hyponyms, coordinate terms etc. (WordNet, 2016). The popularity of the English WordNet and the fact that it is popular with NLP research, makes it an ideal tool when building software made to do NLP calculations. An example of such a software is the Natural Language Toolkit (2016). Built in python, and has a wide array of functions, using lexical resources like WordNet. Perl modules built by Ted Pedersen et al. SenseRelate modules are another popular example of software built by using WordNet. I believe that the use of such tools will be vital for the project.

There are several semantic similarity measures available within the vicinity of WordNet. In both the Test & Results section, there is some presented with a short explanation. The measures presented are the ones available with Perl, through the WordNet::Similarity module.

4.3.1 USING WORDNET

WordNet, as mentioned above is an interlinked database of synsets that are grouped together by their meanings. Using this tool in a project that is comprised of disambiguating words, seemed like the logical thing to do. A substantial amount of work around the WordNet database has also been done; be it Markov models, Disambiguation software, or Part-Of-Speech programs. Having in mind this, locating the relevant programs usable to me in the

project was a good place to start as any. For simple testing and researching words and synsets within WordNet, WordNet online search (WordNet Stanford) suffices. If the online version should not be available, a downloadable version is available. From the same domain.

For programming and development use, several different WordNet access software are available. As an example NLTK for python development grants WordNet access and the WordNet::QueryData sub module, grants access into the database. There are WordNet libraries available for most known programming languages, finding them for the research is a matter of searching. Nonetheless starting with the Perl module SenseRelate which is combined with the access point of QueryData will be one of the priorities to explore.

4.3.2 TECHNIQUES

WordNet::Similarity is a Perl module which includes Semantic relatedness techniques. Semantic relatedness measures how alike two terms are with one another. An example of a similarity measure is between "dog" and "cat", they are more similar than "dog" and "car", but "dog" is also related to "bark" and "bite". Not to be confused with measuring the similarity in how words are presented, i.e. their string form. In that case "car" and "cat" would be very much alike.

Using the WordNet::Similarity module I can measure the relatedness between senses, as mentioned before, a word can have several meanings and according to the context, one or more fits better than the rest. How to measure which of these that fits better? Using relatedness measures against the surrounding context is a start.

The measurements available with The Perl WordNet module are presented below:

. (The explanations are from WordNet::Similarity pages from CPAN)

1. Wup Measure: Wu & Palmer (1994), calculates the relatedness between two synsets' depth in WordNet. The score from the algorithm I between zero and one. One if the synsets are the same.
2. Res Measure: (Resnik 1995), measures relatedness by measuring the information available from concepts.
3. Random Measure: Measures using a random similarity measure.
4. Path Measure: Measure by counting nodes in the Wordnet 'is-a' hierarchy.

5. Lin measure (Lin 1998), measures by the content of information to get a similarity measure between synsets.
6. Lesk Measure (Banerjee and Pedersen 2002): Lesk method that measures senses by glossary overlaps.
7. Lch Measure: (Leacock, Miller, and Chodorow 1998), Counts the edges between senses in a 'is-a' hierarchy. The value is normalized by the maximum depth of WordNet, followed by a negative log of the normalized value.
8. Hso Measure: Hirst & St-onge(1998), method of identifying lexical chains in text as described in their paper.

Of the mentioned measures mentioned above, the accepted POS pairs are:

1. Wup measure: [['n', 'n'], ['v', 'v']]
2. Res measure: [['n', 'n'], ['v', 'v']]
3. Random measure: **NR**
4. Path measure: [['n', 'n'], ['v', 'v']]
5. Lin measure: [['n', 'n'], ['v', 'v']]
6. Lesk measure: [['a', 'a'], ['a', 'r'], ['a', 'n'], ['a', 'v'], ['r', 'a'], ['r', 'r'], ['r', 'n'], ['r', 'v'], ['n', 'a'], ['n', 'r'], ['n', 'n'], ['n', 'v'], ['v', 'a'], ['v', 'r'], ['v', 'n'], ['v', 'v']]
7. Lch measure: [['n', 'n'], ['v', 'v']]
8. Hso measure: [['a', 'a'], ['a', 'r'], ['a', 'n'], ['a', 'v'], ['r', 'a'], ['r', 'r'], ['r', 'n'], ['r', 'v'], ['n', 'a'], ['n', 'r'], ['n', 'n'], ['n', 'v'], ['v', 'a'], ['v', 'r'], ['v', 'n'], ['v', 'v']]

These methods are tested in a small scale and presented in the Development section. Having an abundance on semantic similarity measures creates ground for testing which would be the better choice. Using the optimal measurement in the development section, will decrease the amount of time it would take to build algorithms able to solve the research question. Developing own algorithms to measure the similarity would perhaps be the most thorough way. Nonetheless, the time available will not allow for such development, and those steps does not help in answering the thesis question any more than using previous measurement methods would.

5 DEVELOPMENT

This section will present the stages in developing the WSD software. The stages include figures and reasoning behind the implementation. Figures are code sections with explanatory text. The programs presented below all deliver to a degree correct senses when run with a target word and a context. In order, the pre-existing algorithms are presented, and secondly are the methods proposed in this thesis. Testing and examining the pre-existing algorithms before developing the thesis based algorithms is key for understanding where to improve or what functions to add. It is important to note that all of the algorithms, either pre-existing or not, are based completely on WordNet.

5.1 Stage 1: SenseRelate

This is the first stage of the development; this means that some time was used to find software that could be reused and examined before my own. The natural course of action was to see what WSD software there was available for WordNet. First on the list of software to start with is the SenseRelate modules. Developed by Pedersen et al., the development was based on a previous method, and built the SenseRelate package further over the years. The main modules in question are WordNet::SenseRelate::AllWords, TargetWord, and WordToSet. A short introduction of them and a code example of the base usage followed by how they were first implemented in the program.

5.1.1 WORDNET::SENSERELATE::ALLWORDS

Pedersen is the main developer in building the SenseRelate, this current module which will be called AllWords from now on, takes in the base case a set of words(context) and assigns the most probable sense to the words based in the given context. Adding to that, there is the availability to tweak the module by choosing which kind of similarity measure (see section 7.3.2) the algorithm is going to use, and whether it is going to focus on the Nearest word when choosing senses or from the whole set(globally). This Module differs from the others in that it assigns a sense to every word in the context given. The exception are words that is not listed in the WordNet lexicon.

Code example below:

```
my $wn = WordNet::QueryData->new(
    dir     => "D:/Perl/wn3.1.dict.tar/wn3.1.dict/dict",
    verbose => 0,
    noload  => 0
    );
my $wntools  = WordNet::Tools->new($wn);
my %optionsa = (
    wordnet => $wn,
    wntools => $wntools,
    measure => 'WordNet::Similarity::lesk'
    );
my $obj = WordNet::SenseRelate::AllWords->new(%optionsa);
```

Figure 5-1. All-Words Setup

```
my @context = split / /, $x;
my @res           = $obj->disambiguate(
    window  => 3,
    scheme  => 'normal',
    tagged  => 0,
    context => [@context]
    );
```

Figure 5-2. All-Words example

Figure 2. Shows the setup of the module. Where $wn is the WordNet object, the *%optionsa* hash list adds the WordNet object and selects the measurement technique which in this case is the Lesk similarity measurement (Banerjee and Pedersen 2002). And finally initializing the AllWords module on the last line with the *%optionsa* argument along.

//my @res = $obj->disambiguate (...) is where the actual command to disambiguate is. Within the parentheses are the arguments. The important argument is the *@context* which a list of words to be disambiguated. I have added the "split" line in the example to show what format the disambiguator needs. The split function in Perl divides a String into a list. Splitting is done on what you give it, and in this case, on whitespaces.

Since the AllWords disambiguation module attaches a meaning or sense to every word in the context, a test run of the program would put all of the word senses in the same sense group. For example, if the context contained a sentence such as: "She saw the man with binoculars", it would give each of the words (if available) a sense like this: [saw->see#v#1, man->man#n#1, binoculars->binoculars#n#1]. This is but one example, it would work on longer sentences as well as short ones.

5.1.2 WORDNET::SENSERELATE::TARGETWORD

This module disambiguates a target word instead of the whole context like explained in the previous SenseRelate module. The TargetWord module has a bit more setup than the others, but works in a similar way with the exception of telling the module which word you want disambiguated. The idea of the module was to extend beyond glossary overlaps like (Lesk 1986), TargetWord implements the functionality of finding the sense that is most related to its neighbouring senses. The measurements are specified by the user (see Figure 8- 4).

The setup and execution is shown below:

```
my $wn = WordNet::QueryData->new(
  dir     => "D:/Perl/wn3.1.dict.tar/wn3.1.dict/dict",
  verbose => 0,
  noload  => 0
  );

my %wsd_options = (preprocess => [],
 preprocessconfig => [],
 context => 'WordNet::SenseRelate::Context::NearestWords',
 contextconfig => {(windowsize => 5,
   contextpos => 'n')},
 algorithm => 'WordNet::SenseRelate::Algorithm::Local',
 algorithmconfig => {(measure => 'WordNet::Similarity::res')});

  # Initialize the object
  my ($wsd) = WordNet::SenseRelate::TargetWord->new(\%wsd_options, 0);
```

Figure 5-3. TargetWord module setup

```
my @splitContArray = split / /, $context;

foreach my $theword (@splitContArray)
{
  my $wordobj = WordNet::SenseRelate::Word->new($theword);
  push(@{$hashRef->{wordobjects}}, $wordobj);
  push(@{$hashRef->{words}}, $theword);
}
my ($targetIndex)= grep { $splitContArray[$_] eq $targetWord } 0..$#splitContArray;

$hashRef->{target} = $targetIndex;
$hashRef->{id} = "Instance1";

my ($sense,$error) = $wsd->disambiguate($hashRef);
```

Figure 5-4.TargetWord Execution

From *Figure 8-5*, there is a bit more setup than on the AllWords module. Choosing how to disambiguate from the context, the similarity measurement algorithm, to the context part of speech type. Using the module "out of box" works better than tweaking the module, it tends to crash and seldom work with other combinations.

The module needs to have the context converted to "SenseRelate::Word" objects before it can disambiguate. *//my ($targetIndex) grep {...}* ... is the line where the program extracts the position of the selected word, which is then added to the *$hashRef* hash list which contains the options that the disambiguate function needs. The result from the disambiguation will return a lot of warnings from the Perl interpreter when run with the *strict* and *warnings* pragmas. However, it does return a disambiguation from the run, not always correct, but that will be presented in the results section.

5.1.3 WORDNET::SENSERELATE::WORDTOSET

WordToSet is the last of the SenseRelate modules that has been used in the thesis. This module like it says in the name takes a set of context words and a word that is to be disambiguated. Above, the implementations require an index on which of the words one want disambiguated. For example, AllWords disambiguates everything, so retrieving the target disambiguation requires an index. TargetWord disambiguates the word in a sentence array based on the index of that array. So if the target word is placed in index number three if the array, then one disambiguate like so: sentence[index]. This is different in WordToSet in that the requirement is that the target word has to be within the context. See *Figure 8-7* in the line where *//my $res = $mod->disambiguate (...)*, the "*$word*" has to be within the *@sentenceArray*, if not, the program crashes.

Examples below:

```
my $qd = WordNet::QueryData->new;

my %options = (
    measure=>'WordNet::Similarity::lesk',
    wordnet=> $qd);

my $mod = WordNet::SenseRelate::WordToSet->new(%options);
```

Figure 5-5. WordToSet setup

```
my @sentenceArray = split / /, $sentence;

my $res = $mod->disambiguate(
    target => "$word",
    context => [@sentenceArray]);

my $best;
my @resList1 = ();
my $best_score = -100;
foreach my $key ( keys %$res ) {
    next unless defined $res->{$key};
    if ( $res->{$key} > $best_score ) {
        $best_score = $res->{$key};
        $best       = $key;
    }
}
print "$best : ", join( ", ", $qd->querySense( $best, "glos"));
```

Figure 5-6.WordToSet execution

The setup is quite short in this module, it only requires a WordNet object and a similarity measure, the execution of this module is compared to the others the most troublesome of them all, because it requires some sorting and score measurements in code. In *figure 8-7*. one can see that it runs through the *$res* hash and attempts to find the word with the highest score thus retrieving the sense with the highest probability of being the correct one. The code sorts the senses that has a value attached. The functionality of returning a hash list of senses with an accompanying score, separates WordToSet from the other two SenseRelate algorithms, and the NLTK WSD algorithm.

5.2 Stage 2 NLTK- Natural Language ToolKit

This stage expanded in exploring into another WSD tool, namely, NTLK(NAVARRE and STEIMAN 2002). The toolkit has access to WordNet, which is key to the thesis. This stage has only one software to present, however the amount of time consumed into both learning a new programming language and how to use it properly makes it qualified to be written in its own stage. Accessing the WSD function within NLTK is a simple task (presented below Figure 8). Expanding the toolkits use can be seen in the section 8.3.2, where the Normalised Google Distribution semantic similarity is examined, in conjunction with Python, and pythons NLTK distribution.

5.2.1 NLTK - WORD SENSE DISAMBIGUATION

This subsection presents implementation and a simple run using the NLTK Word Sense Disambiguation.

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from nltk.wsd import lesk
>>> sent = ['I', 'went', 'to', 'the', 'bank', 'to', 'deposit', 'money', '.']
>>> print(lesk(sent,'bank','n'))
Synset('savings_bank.n.02')
>>> print(lesk(sent, 'bank'))
Synset('savings_bank.n.02')
```

Figure 5-7:NLTK wsd Lesk example

As the figure presents above, it is a simple import and run from console. One implements the lesk function from the nltk.wsd module, followed by running the lesk function with the target word and a sentence (The part-of-speech tag is optional). From the import it is reasonable to assume that the WSD algorithm has been built on the Lesk measurement (section 7.3.2). The WSD function is also based on WordNet like the SenseRelate algorithms.

The synsets' definition:

savings bank#2, coin bank#1, money box#1, **bank#8** (a container (usually with a slot in the top) for keeping money at home) *"the coin bank was empty"*.

One of the issues I had with the program software, was that the target word had to be within the context in an identical way for the disambiguation to succeed. For example, if the target word had a plural form and in the context it was not, then it would either not return any sense, or simply end the program with an error code. This is alike to the WordToSet module presented in section 8.3.2.

5.2.2 NGD- NORMALIZED GOOGLE DISTANCE

NGD is a technique to find semantic similarity using the google search engine. The technique calculates the number of results from two search words and the actual number of pages available. The formula is presented below.

$$\mathrm{NGD}(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

Equation 5-1: NGD formula

X and Y are the number of results from each search, and after the subtraction, *(log (f (x, y))* represents a google search with both the words. N is the number of pages available through google. The closer to zero the more similar the word.

The premise is good, and the similarity results (based on simple tests) are not bad, but using the technique efficiently requires that I have activated a premium account that allows me to exceed the 100 daily google query limit. Furthermore, the need for constant internet connection is a necessity for this similarity measure. For a WSD algorithm, several measurements are required, which means the amount of connections and searches will summarize an amount of time that will not be feasible for actual program usage.

The calculation is presented below.

```
def computeNGD(x,y):
  if x == y:
    return 0
    x_ = log(float(google(x)))
    y_ = log(float(google(y)))
    f_xy = log(float(google(x +" "+ y)))
N = 50 * 1e9   # total number of indexed pages
return (max(x_, y_) - f_xy) / (log(N) - min(x_, y_))
```

Figure 5-8:NGD Calculation

The line with the google () method, is the method reference which returns the search results with the search word as the parameter. N = 50 * 1e9 at the time was the number of pages available. The google method activates the google api with the developer key, and as mentioned it returns search result numbers. Below are three measurements with individual search results and combined search results followed by the similarity score:

NGD example:

Similarity: X Vs Y	**NGD**
Dog Vs Cat	0.17533390488765704
Dog Vs Car	0.24571626187298848
Car Vs House	0.09133073825913088

Tabell 5-1: NGD similarity example

5.3 Stage 3 Manual Disambiguation stage

The idea of the manual disambiguation came when researching the Lesk measurement system and its POS abilities in Perl WordNet::Similarity. The idea is to use Lesk (Banerjee and Pedersen 2002) to measure synsets from the context words against the target words senses. Meaning, if the target word has 4 senses, these four senses will be measured against each of the context words synsets (except itself). The measurements are done through algorithms presented below in the following subsections. Using Lesk as the main similarity measurement is motivated by the results and the ability presented in the testing and results section (see section 11.1, and 12.1). The only exception is the first Manual Disambiguation (see section 8.3.1), where Wup (Wu and Palmer 1994a) is used. The reason for this is that the Lesk measurement is not available as a standalone semantic measurement in NLTK.

The manual disambiguation methods below are the new methods proposed to reach the thesis goal. WSD algorithms that return a ranked list of most the most probable senses. The Algorithms are individually built upon one another, meaning ideas and methods developed in the first manual are built upon in the next. Improving the algorithms capabilities over time. The first and second manual algorithm have access to a context and a target word within that context. Creating a more lifelike scenario in the algorithms area of use. The third and fourth have access to the same as one and two with the added heading of a webpage. The idea is that there are strong clues to what information there are in the web sites text in the heading of said web page. For example, if there is a news article with the heading: “Ducks” there is a good chance the webpage is about the ornithological species and not the action of ducking when there is a foreign projectile headed for you. Before the manual algorithms are presented a rationale of POS – Tagger and Semantic Relatedness usage. The subsections explains with preliminary tests and reasoning, why the different functionalities have been used in the Manual algorithms.

5.3.1 SENSERELATE::SIMILARITY TESTING

This subsection presents the testing done to decide which of the semantic similarity measures available in Perl, performed better. Running and testing the SenseRelate algorithms, it occurred to me that the algorithm will behave differently with different semantic similarity algorithms. But How Differently? Which of the measurements should I use? Building a Perl program which ran all of the senses against a collection of contexts and target words allowed me to see the differences in the methods. The collection of target words and context sentences is collected so that there is a definitive word sense to be chosen.

Presented below is the small test collection:

```
"break"=> "The glass broke in a thousand pieces i hate it when things break",
"light"=> "he shead som light on the situation at hand, it was hard to understand",
"charge"=> "The bank charged him for not paying his bills on time, he lacked the funds to do so",
"face" => "Look for the dice and tell me what face it shows",
"cat"=> "It was used on slave trade ships to punish the prisoners by whipping them with the cat",
"dog"=> "That person sure has an ugly face, a real dog",
"stab"=> "He took bullet and stab wounds to the head, face, side, and arm, returning fire the whole time.",
"tank"=> "WW1 tactics and theoreticians soon announced that at least three tank types would be necessary to make a difference in no-man's land
```

Figure 5-9: Word and Sentence collection

The different semantic measurements are as presented in the Techniques subsection; these will be the main methods in question. The preliminary results are in the table below:

MEASURES	Correct	Incorrect	Correct %
WUP	1	7	12,5
VECTOR		8	0
VECTOR PAIR		8	0
RES	2	6	25
RANDOM	2	6	25
PATH	1	7	12,5
LIN	1	7	12,5
LESK	2	6	25
LCH	1	7	12,5
JCN	2	6	25
HSO		8	0

Tabell 5-1: WordToSet Similarity measures result

The First look on the results shows that, words with a great number of senses available have a big difficulty of reaching the correct sense, the granularity of senses is too close to each other for an algorithm to efficiently separate them. This is an example of the difficulties algorithms have when faced with polysemy. An example of such a word is the first one in the example

above. “break”, has seventy-five different senses, sixteen noun senses, and fifty-nine verb senses. That means that the granularity of the senses, makes it difficult for humans to distinguish the correct sense. Following, a WSD algorithm will have an equal if not increased trouble of choosing the correct one. One solution could be to have more context, and some way of extracting nouns and verbs that can help the disambiguation algorithm to differentiate between senses.

The measurement takes a while, and viewing the runtime of the semantic measurements it is clear that Hso is the time consuming culprit. Presented below are three sample runs using WordToSet with Hso as the semantic relatedness measurement.

First run:

Sentence: “The dog fought with teeth, fangs, and claws".

Target word: “dog”. (Number of senses: 8)

Word Sense Disambiguation Results: ***dog#n#1***: a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds; "the dog barked all night"

Time used: 59.75 seconds.

Second run:

Sentence: “A male duck is called a drake and the female duck is called a duck, or in ornithology a hen”.

Target word: “hen”. (Number of senses: 4)

Word Sense Disambiguation Results: ***hen#n#3***: flesh of an older chicken suitable for stewing

Time used: 3 minutes and 19.3 seconds. (193.90 seconds)

Third run:

Sentence: “To be able to submerge more easily, the diving ducks are heavier than dabbling ducks, and therefore have more difficulty taking off to fly”.

Target word: “fly”. (Number of senses: 20)

Word Sense Disambiguation Results: ***fly#v#7***: change quickly from one emotional state to another; "fly into a rage".

Time used: 19 minutes and 22 seconds (1161.97 seconds)

The results from table 1 was below expectations, under 50% correct, going into the experiments the hopes were in the seventy percent vicinity. The measures were as mentioned done with the WordToSet module, this could be the reason for the low score, and the size is likely to be a factor. At this time this is the quickest way to see which of the similarity measures to use in development. From table 1, Resnick, Lesk, and JCN scored correct 25 percent of the time. This combined with Lesk' ability to measure against 100 percent of the synsets in WordNet, supports the choice in using Lesk as the main semantic measurement. It can be argued that the test size is not exhaustive enough to return a significant enough result, but given that measuring the competence of semantic measures is not a focus in the thesis, it would have to suffice. Observing table 1, one can see that Vector and Vector Pair are tested, the two are not mentioned in section 7.4.2. They are not used in any of the development, and is tested in this section for completion. This early preliminary test proved this phase, in the way that the two synset relatedness measures did poorly in conjunction with the WordToSet module.

Hso,(Hirst and St-Onge 1998) is the only other measurement that accepts multiple POS elements and is cause for the timed run shown below table 1. Since the run time in the three tests greatly increases with the size of the context it would be fair to assume that an increase in context words would increase run time. It would also seem like the number of senses also have a correlation with run time as well. Using an algorithm that takes up to a minute of runtime, even if the disambiguation corresponds to the context, is not functional.

The results from the measurement and time test of the Hso measure combined with the POS functionality, shows that the WordNet::Similarity::Lesk measure is the optimal choice to use in the development.

It is important to note that in this context, i.e. WSD within the Websites context, that even though the Lesk measure is the measurement choice in measuring the word sense in every POS element. It does not mean that it is the optimal choice for measuring for example exclusively Nouns or Verbs which the other algorithms do, such as Lch and Res measurement (section 7.3.2). If that is the case, then further research is required. But for this thesis, the task dependent word sense disambiguation algorithms built in this thesis, Lesk is the correct choice for semantic measurement.

Possible Issues

This test was run alongside the SenseRelate::WordToSet module, and the module accepts all of the semantic similarity measures. However it is the Lesk(Banerjee and Pedersen 2002) algorithm that accepts multiple part of speech elements to be measured against different elements.

The issue with using the Lesk algorithm, is that the other algorithms might do a better measuring job than the Lesk algorithm, and that the quick test-run done on the measurement is not enough to decide on which measurement that does the best job.

5.3.2 POS – TAGGER RATIONALE

The first of the integration of the pre-existing algorithms, it became clear that the Perl, and NLTK word sense disambiguation algorithms did not allow for other part of speech disambiguation other than nouns and verbs (only reflexive). This thesis has focused its work using WordNet as the English corpus of use, and in this corpus there are a total of 155287 unique strings divided over four Part of Speech elements(Nouns, Verbs, Adjectives, and Adverbs)("WordNet Stats 2016-11-23" 2016). Of this total 75,86 percent are nouns. Verbs account for 7,4 percent. In total this accounts for 83,28 percent of unique words in WordNet, and explains why Nouns and verbs are the main concern in WSD within WordNet.

It still leaves 16,72 percent of possible words divided over Adverbs and Adjectives. From the results section one can review the number of senses retrieved from each of the algorithms, and the results correlate to the numbers shown in this section (with the exception of WordToSet algorithm), NLTK, AllWords, and TargetWord which missed respectively 18,15, and 16 which averages to 16,33 percent, which is in range with the missing adverb and adjectives from the total. Using a POS-tagger in a web application where humans are to choose a sense for a word would exclude a percentage of senses which could be mistaken for another part of speech element. This goes for the algorithms as well. The other aspect of this is the possibility of error in using POS-tagger algorithms, i.e. mistake a noun for an adjective etc. This would exclude the correct meaning of the target word. This error is expressed in the theory regarding to POS taggers. The correct tagging depends highly on correct sentence builds and spelling, which could prove to be a problem with online text, since there seldom are spell checking for blogs, social media posts, online articles etc. Even though there is a risk of eliminating the

most sensible sense from the list of senses. Including 100 percent of the words in WordNet and the ability measure relatedness between them outweighs this risk.

5.3.3 MANUAL DISAMBIGUATION I

The first Manual Disambiguation method runs a POS software on the context and finds the POS element of the target word. Excluding the meanings that is in the other POS elements, following this, the algorithm loops through every one of the target senses, measuring them against all possible synsets from the words in the context, the measurement score is added to the target synset for every measure. The synset with the highest score is the most probable sense. The result is a hash array with the target senses as key, and the accumulated measurement score as the value. This stage of the manual disambiguation was written in python, and used the NLTK to access the WordNet Synsets.

This stage of the disambiguation limits the part of speech tags to Nouns and Verbs. The reason for that is the available measures in the NLTK either do verb and verb or noun and noun. At this stage the POS elements are limited to Nouns and Verbs, this means that when given a sentence, everything but the nouns and verbs are cleared so as to not crash the program mid execution. Presented below is the measurement part of the program:

```
measure = 0
measure = targetSS.wup_similarity(wordSynset)
#Add the measure to the list.
synsetHashValues[targetSS] += measure
print("********** VALUE ************       :     ", synsetHashValues[targetSS])
print("Measure: "+str(wordSynset)+" : "+str(targetSS)+" = "+str(measure))
```

Figure 5-10: Measurement accumulation between synsets

SS is just an abbreviation for synset and not for the “Schutzstaffel” from world war II. In the second line from the top the measurement is made with WUP similarity(Wu and Palmer 1994b), below the comment is the addition to the hash list.

The end goal of this method was to end up with a ranked list of the targets senses, and on top of the list will be the most likely sense of the word according to the context. The negative elements to the program is that it is limited to nouns and verbs, when WordNet has more Part of speech members than nouns and verb. That said, for the nouns and verbs, results where promising from single sentence context. The similarity measures available in the NLTK module is limited in comparison to what the Perl SenseRelate module can offer, which is what will be used in the following section.

5.3.4 MANUAL DISAMBIGUATION II

The second part of the manual disambiguation are developed in Perl, making use of the techniques from section 7.3.2. Like the previous section this is based on the Manual algorithm technique, this part of the development uses Lesk as the primary synset measurement technique, the rationale is shown in section 8.1. As mentioned the, acceptable POS pairs are: [['a', 'a'], ['a', 'r'], ['a', 'n'], ['a', 'v'], ['r', 'a'], ['r', 'r'], ['r', 'n'], ['r', 'v'], ['n', 'a'], ['n', 'r'], ['n', 'n'], ['n', 'v'], ['v', 'a'], ['v', 'r'], ['v', 'n'], ['v', 'v']]

(r in this context, or rather, WordNet's context, is an adverb). Instead of just [[v, v] [n, n]]

So running the program proceeds as following:

1. Find the target words part of speech in the context and retrieve its POS senses.
2. For every target sense, loop over the contexts word, also retrieving the senses from that word (its POS senses)
3. Measure the current target word against the context words senses and add the score to a <key, Value> list (hashmap for java users) where Key is the targets senses, and value is the similarity score.
4. Print the sorted result to review which of the target senses which has the most amount of points.

A sample output will be presented below, after a small code snippet showing the particular disambiguation. This solution to a possible disambiguation was motivated by the previous idea on "manual disambiguation", being able to disambiguate against more than only nouns and verbs, presents more work for the algorithm, but at least it includes possible correct word meanings.

And as pointed out above, using a POS tagger, one excludes in some instances, 1/4 of available senses not applicable to the context.

Below is a code Snippet showing the disambiguation:

```
#the loop over the contextword' senses
foreach my $contSense (@contSenses) {
    if ( $contSense eq $targetSense ) {
        next;
    }
    my $target_Cont_measure
    = $measure->getRelatedness( $targetSense,
        $contSense );

    $contSenseHash{$contSense} = $target_Cont_measure;
    $valueHash{$targetSense} += $target_Cont_measure;
```

Figure 12. Context sense loop and measurement

The measurement is calculated through the *->getRelatedness* method, and the score is added both to *$contSenseHash* and the *$valueHash*. The *$contSenseHash* is for presentation and debugging purposes, the $valueHash is sorted and includes the final results. The important part is that the measurement of the target words sense and contexts sense is added to the overall measure of the target synset.

The sorted results of the targets synset is the final disambiguation.

```
dog#n#5 -- value : 349
########DEF########
a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll
---
Synset: frank#n#2, frankfurter#n#1, hotdog#n#3, hot_dog#n#3, dog#n#5, wiener#n#2, wienerwurst#n#1, weenie#n#1
###################
dog#n#6 -- value : 682
########DEF########
a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward
---
Synset: pawl#n#1, detent#n#1, click#n#3, dog#n#6
###################
dog#n#1 -- value : 1263
########DEF########
a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times;
all night"
---
Synset: dog#n#1, domestic_dog#n#1, Canis_familiaris#n#1
###################
```

Figure 13. Manual disambiguation results

The first noun sense is the one that is chosen according to this context, this is often the result of disambiguation, i.e. defaulting to the first sense of the POS tag. More on the results from the different stages on the result section. The big difference in this method over the previous one is being able to measure against different POS tags.

Trouble

Though using POS taggers can help an algorithm choose the correct sense more often, a POS program can tag the sentence wrong. And in choosing the wrong sentence element for disambiguation, it would exclude the correct sense. It could be the case that the granularity between the senses are so fine that it could be cross-POS similar senses. Still, it would not be the correct POS or sense.

5.3.5 MANUAL DISAMBIGUATION III

This stage is an extension of the previous disambiguation stage, it measures and adds similarity points the same way. The difference is that it includes a score multiplier when applied to a websites headline i.e. a heading in an HTML document. The premise is that if the target word was included in the title in a way, then the disambiguation should weigh more. Following that idea, Adding the function so that if any of the words synset is equal to any of the words in the title, then, any measurement using the particular synset got an added weight to the score. Example: searching WordNet for the word "cat", the 7th sense of the word has the synset: "big cat". So if the title sounds like this: "Victim killed in big cat attack". The 7th sense should then get a multiplier when measuring against the other senses, and the other measured in a normal way, unless they also have a synset included in the title. Further work could be to add some fuzzy logic on how closely similar they are would decide how big the multiplier should be.

Below is a small sample of how the measurement is done.

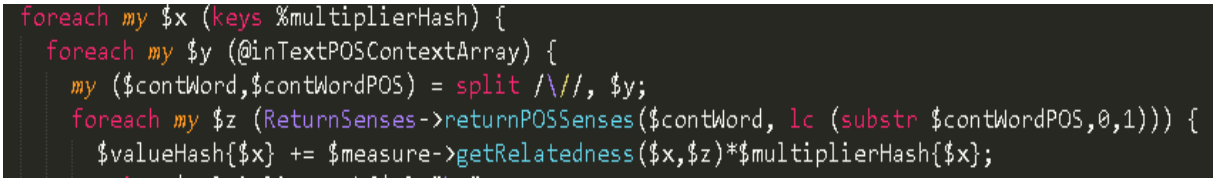

Figure 14: Synset multiplication

The fifth line in the image is where the multiplication happens. Previously, the algorithm checks if the different synsets from the target senses is within the headline as explained above. Further work on this algorithm could expand to the in-text-context as well, maybe giving the synsets of the context words a score multiplier when the word is connected somehow to the headline. So, when the measuring starts the target senses will also be multiplied when measured by some of the senses within the context.

5.3.6 MANUAL DISAMBIGUATION IV

Following the success in being able to use several POS tags in disambiguation, this motivated me to expand deeper into what WordNet module has to offer in terms of glossary, using the definition of each synset that the target word has, and then measuring the relatedness score against the context word synsets. This stage is purely experimental, however, the algorithm used to accomplish this stage is complex, (and perhaps unnecessary), but no stone shall remain unturned. The reasoning behind such an algorithm is to use the "Manual Disambiguation" idea on the content available from WordNet, and by content it is meant by the glossary from each sense. The glossary is meant as a definition/explanation on the synset. So, if the sentence is a representation of the synset, it stands to believe that the content is highly related to the synset. Following that reasoning, measuring that content against the context we want to find the correct sense in, can bypass the granularity problems often seen in WordNet synsets. Below is the implementation of the algorithm included with explaining comments.

```
sub GlossaryDisambiguation{
  my ($target,@cont) = @_;
  my %subValueHash =  ();
  #Loop: 0, Runs through each of the target words available senses
  foreach my $x (ReturnSenses->returnPOSSenses($target,$targetPartOfSpeech)) {
    print "$x\n";
    $subValueHash{$x} = 0;
    my @glossPOSArray = split / /,(lc $p->get_readable(($wn->querySense($x,"glos"))));
    print @glossPOSArray;
    #Loop: 1, Runs through the current target senses pos tagged definition
    foreach my $y (@glossPOSArray) {
      print "$y\n";
      my ($glossW,$glossWPOS) = split /\//, $y;
      #Loop 2: Runs through each of the senses for each of the words in the definition
      foreach my $z (ReturnSenses->returnPOSSenses($glossW,(lc substr $glossWPOS,0,1))) {
        print "$z\n";
        #Loop 3: Runs through each of the context words
        foreach my $xy (@cont) {
          my ($contextW, $contextWPOS) = split /\//, $xy;
          #Loop 4: Runs through each of the context words senses
          foreach my $xyz (ReturnSenses->returnSenses($contextW)) {
            print "$z----------------$xyz\n";
            #measuring target words definition word against context word senses.
            $subValueHash{$x} += $measure->getRelatedness($xyz,$z)*$multiplier;
            print $measure->getRelatedness($xyz,$z),"\n";
```

Figure 15. Glossary Definition algorithm

The loops are five level deep, this means that it takes an amount of time and memory to run. The need to reach that deep is necessary:

1. Running through the senses of the target word.
2. Running through each word in the glossary
3. Running through each sense of the glossary
4. Running through the context words
5. Running through the context words senses.

Finally, measuring the senses and accumulating the score between the glossary word sense and the target word sense. Since, this algorithm takes quite the amount of time and power, it is less likely that this method will be used in an actual WSD software. However, it is interesting to see the results from the use of glossary content against the sentence context. Results of the tests are presented in the Results section.

The method is run against a websites headline and a sample context. In the same way as in the previous section, if the actual word, valid form of a word, or a synset is included in HTML heading, then the measurement is complemented with a multiplier.

6 TESTING

This section will present how the testing phase of the algorithms built during the thesis was done. The important part of this section is the algorithm testing against humans. First, the pilot human-software test. Complete with the initial Algorithm built through the thesis run against the results from the Human part of the investigation. Secondly presented is the main experiment, both the human side of the tests and the algorithms.

6.1 Pilot Experiment

The initial testing against humans was initiated after the first multi POS algorithm was built. With the intent to see if the program chose the same answers as a human would. The test was comprised of fifteen different multiple choice disambiguation tasks, executed in paper form. Each one has a headline, a context sentence, and a word to disambiguate. Below the lines of context and target word, are several lines of senses that the user could choose from. These senses were divided by the words available POS tags. This way, the human subjects are subjected to the same level as the disambiguator. The test was given to three different candidates who agreed to participate in my experiment. The amount could have been greater, but seeing this as a preliminary test to the big one, using five or less people should suffice. It is important to mention that two out of three of the participants have Norwegian as their native language and English as the second language.

An example of the test is shown below:

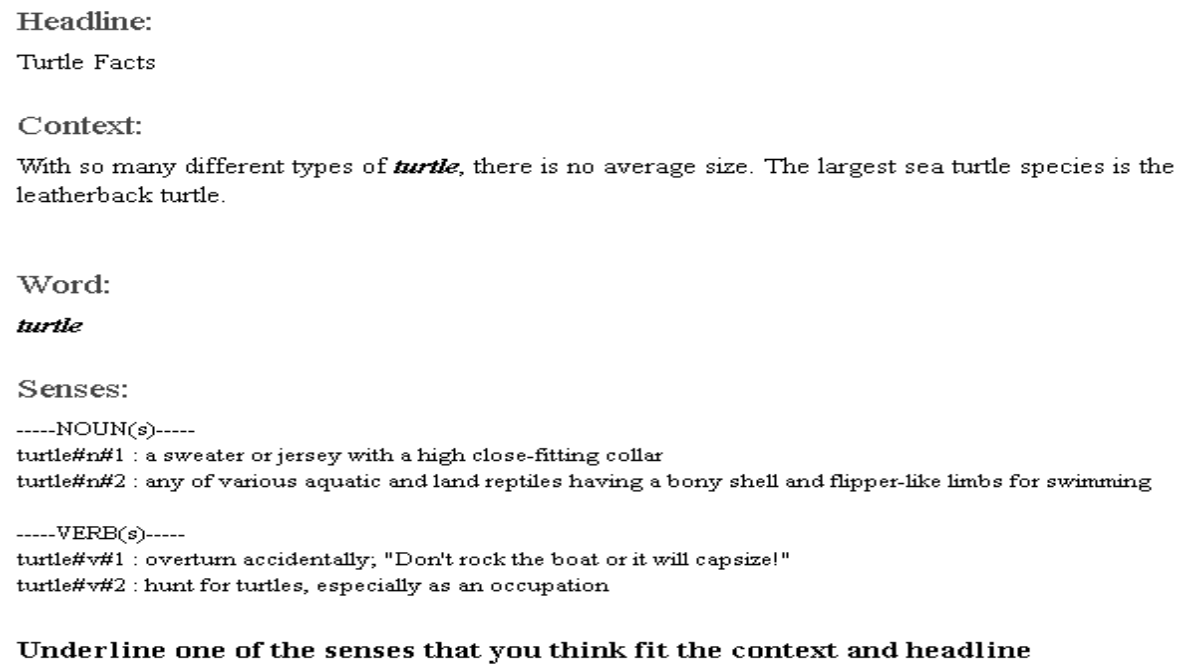

Figure 6-1: Pilot experiment question sample

This is one of the fifteen different tasks given to the humans. The only thing the subject needs to do is to underline or mark the sense which makes most sense to him or her.

The same data will be used on Manual1. Since the test data is of a relatively small size, they have been run individually with the algorithm. That is, not run them automatically, this is possible with the small amount of data in this test phase.

The resulting answers from the three raters are presented below with the accompanying Manual1 results.

rater 1	rater 2	rater 3	program
underground#r#2	underground#a#1	underground#a#1	underground#a#2
wound#n#2	wound#n#1	wound#n#2	wound#n#1
bomb#n#1	bomb#n#1	bomb#n#1	bomb#n#1
turtle#n#2	turtle#n2	turtle#n#2	turtle#n#2
chair#n#1	chair#n#1	chair#n#1	chair#n#1
throw#n#1	throw#v#1	throw#v#1	throw#v#1
drive#v#1	drive#v1	drive#v#1	drive#v#3
dig#v#1	dig#v#1	Dig#v#1	dig#v#1
jump#v#10	jump#v#8	Jump#v#10	jump#v#1
aim#n#3	aim#v#1	Aim#v#1	aim#v#1
small#a#1	small#a#1	Small#a#3	small#a#1
full#a#1	full#a#1	Full#a#1	full#a#1
free#v#1	free#v#3	Free#v#1	free#a#1
safe#a#1	safe#a#1	Safe#a#1	safe#a#1
wrong#a#1	wrong#a#1	Wrong#a#3	wrong#a#2

Table 6-1: Pilot human WSD study

This pilot expresses the difficulty in agreeing on a sense. Deciding on the POS element is also an element of disagreement. Looking closer to the choices of the users, there is a disagreement on the what POS element the word is. The algorithm has a Part of Speech software to help with determining what sentence element it is, and the users have to decide for themselves which sense to use, presented earlier, it is said that the error rate of a POS tagger are quite high when presented with natural text. However, it seems that the disambiguators does this better than humans. Some of the questionnaire answers have senses with fine granularity. For example: jump#v#8 and jump#v#10, as seen on the ninth question. Definition and accompanying synsets below:

jump#8, leap#3, jump off#2 (jump down from an elevated point) *"the parachutist didn't want to jump"; "every year, hundreds of people jump off the Golden Gate bridge"; "the widow leapt into the funeral pyre"*

And:

chute#1, parachute#1, **jump#10** (jump from an airplane and descend with a parachute)

Both can be defined as jumping in a parachute. However, the second sense is more specific in the fact that the synset is definitive parachuting from a plane. Whereas the other can include the action of jumping from an airplane with the accompanying parachute. The disambiguator chose the first sense, which is defined as "move forward by leaps and bounds". Which is not wrong since it makes sense in that context, but it is a to general meaning, and it relates to the idiom. The most accurate one based on the context would be the tenth sense, since the explanation and synsets both point to jumping from a plane with the intent of skydiving with a parachute.

It is interesting how most of the results are defaulted to the first sense of the sentence element. That beckons the question: "How big of a percentage will the algorithms or people default to the first sense of a word according to a context?". The second Human-Computer tests, which includes a drastic increase in participants and questions, proves this.

Issues

Some issues that came along in hindsight of performing the test was:

1. Having the test done on paper, required some time to arrange and make them ready for the test. Having them on paper also increases the chance of the paper getting ruined or messing up the order of the test. Manually reviewing the answers to record them takes some time as well.
2. If the users had the chance to see each of the senses synsets, it would be easier to pick and choose the correct sense.

For the main experiment, these issues will be taken into account when performing the test on a testing software, instead of doing them on paper. Also adding the synsets to the choices.

6.2 The Experiment

To see if I have been able to answer the research question, I have to do another human test phase. So the next step requires some accumulating of representable context and target words within websites. By representable I mean examining websites that for sentences that can give an indication on what the website is about. As an example, the sentence: "A male duck is called a drake and the female duck is called a duck, or in ornithology a hen" and as the target word: "hen". The word "hen" makes sense in the context, but can mean different things based on just the word. When the website is about ducks, both the sentence and the target word is representable. Avoiding non representable sentences and non-descriptive words when running the tests is vital to see if the disambiguator makes the same choices as a human would.

The tests are going to find whether the algorithms agree with humans. And doing a short test as I have done on the available measures is not enough to grant a statistical definitive answer. Having accumulated one hundred different Target-Context test lines. Five different sentences from different websites, that accumulates to twenty different websites with five sentences each. All of them different in the amount of senses and target word usage within the context. Below, there is presented an example of the test lines, this is meant to show how the tests are built can give an understanding of what the tests are comprised of.

```
https://en.wikipedia.org/wiki/Duck|hen|A male duck is called a drake and the female duck is called a duck, or in ornithology a hen
https://en.wikipedia.org/wiki/Duck|fly|To be able to submerge more easily, the diving ducks are heavier than dabbling ducks, and therefore have more difficulty taking off to fly.
https://en.wikipedia.org/wiki/Duck|breeding|Ducks also tend to make a nest before breeding, and after hatching to lead their ducklings to water.
https://en.wikipedia.org/wiki/Duck|quack|A common urban legend claims that duck quacks do not echo
https://en.wikipedia.org/wiki/Duck|domestic|Almost all the varieties of domestic ducks are descended from the mallard (Anas platyrhynchos)
http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|vicious|We tend to think of pirates as bloodthirsty thieves, brutal rapists, and vicious murderers,
http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|shooting|Beyond that, when a typical workday consists of shooting terrified sailors in the face while rival pirates hurl themselves at you with blood-tarnished daggers clenched between their teeth,
http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|defeat|So, alongside an English pirate captain named John Ward, Danziger showed the Algerians both how to build European style ships and how to defeat them in combat, allowing Algiers to extend its piratical reach all over Europe.
http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|career|piracy was an attractively viable career choice.
http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|royal|Actually, Jack Ward, a wildly successful pirate captain who deserted from the Royal Navy along with 30 of his shipmates (because high-seas robbery is way more lucrative than being in the military)
```

Figure 17: Ten test examples

As one can see from the example above, the URLs, target word, and context are all split by the character "|". This is for the automatic testing algorithm built, splitting them this way helps to generalize the building of test suites. The reason for building an automated one, is so that I do not have to run one test at a time. The downside with building such automated tests,

is that, for some of the algorithms which are built differently, devising test suites for most of them takes time. However, even though I have to build individual ones for each one, it will still take less time than running one test at a time.

There are of course the algorithms that require the test websites headline as well as a context and target word combination. The important part of that part of testing these ones is to use the previous test data and just add the headline of the website I got the test data from. The difference between the ones are the adding of the headline. The headlines of websites contain often no more than single word or phrase sentences. For example, the English Wikipedia entry on the "coluber" is comprised of only one word: Cobra. So the difference between the data that contains headlines and those that do not, should not be considered so different that the results could not be compared. On the other hand, the algorithms that takes in the extra context in form of a headline, are different from the others. The question remains if I should do two human trials instead of one. Since the only difference is the extra set of context, the decision to have the users do the test based on the latter test data instead of the first one, landed with the following reasoning: Instead of depriving the human subjects of the extra content to help them differentiate the senses, it may help to do the test more lifelike in the way of presenting the headline as well. After all, if one were to enter a website and browse from top to bottom, I would most likely notice the headline before noticing any of the texts in the different HTML elements within the website.

6.3 The Experiment Questionnaire

The human side of the testing will decide on which of the experimental algorithms built in this thesis that does the best, or decide if they work at all. The results will either way decide the proficiency of the algorithm and maybe see if the context is enough to answer the research question.

The way the testing will go is in the line of how the algorithms were tested, the participant will answer all of the questions linearly, where the sections are divided by headings. In each of the sections there are five different sentences with the accompanying target word. The answers provided is the available senses from WordNet correlated to the target word. To secure that the test goes as planned I will be (as often as possible) situated within the vicinity of the subjects doing the tests. Not expecting anything to go wrong but staying within proximity of the test subject, I can provide with information if there is confusion or if something technical goes wrong during the test. The number of participants of this test should

be around 20 people(G 2016), Reaching a test subject number as high as 20 people can be difficult, since it is a comprehensive time consuming test. Just as the pilot experiment, every participant has English as their second language.

Possible Issues

Taking into account that the programs can fail when retrieving the words from WordNet and that some of the concepts in the WordNet lexicology are more than just single word entries. Such as the combination: "Attack dog". In some of the algorithms, this will be more than often ignored. On the other hand, if the combination is delivered as the target of the disambiguation, the disambiguation works fine. Other problems with the testing section involves the size of the test, people can have experience issues with a test this size. However, having more time to develop the tests nature, so that more disambiguations could be done, could solidify the validity of the results. More on this in the Future Work Section.

Also mentioned above, reaching twenty persons to participate in the testing can prove troublesome. Since the questionnaire is quite lengthy and time consuming, there is a chance that people can lose concentration and the ability to distinguish the senses properly, and end up in choosing randomly. A possible issue for the participants is the fact that; not having English as the first language, could diminish the understanding of the context or the answers.

7 STATISTICS

This subsection will consider the statistics measurement, i.e. the main measurements done to interpret the data collected from the testing phases. The data in question are presented in the Results section (see section 11).

7.1 IRR- Inter Rater Reliability

IRR or inter rater Reliability are a statistics measurement for agreement among raters, or consensus in the ratings given by the raters. Using this kind of measurement for the user and algorithm data makes sense in that I need to find whether the choices humans make are approximately the same as the algorithms. The resulting agreement will help choose the optimal algorithm for the application. Below is presented the IRR measurements that will be used to interpret the Results.

Cohen's Kappa

Cohen's Kappa agreement measure takes two raters categorical data and returns the agreement value between the data sets. The measurement equation is presented below:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

Equation 7-1:J. Cohen Kappa equation (Cohen 1960)

The difference between a general agree percentage is that it accounts for the possibility of agreements occurring by chance. (Cohen 1960).

Below one can see how the kappa measurement can be interpreted, the interpretation from Landis and Koch (1977), is one way to interpret the data, there are more. But the general idea is to use this kind of interpretation to measurements done in this thesis.

Kappa	Interpretation
< 0	Poor agreement
0.0 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 1.00	Almost perfect agreement

Figure 7-1: Kappa interpretation (Landis and Koch 1977)

Basically, if the Kappa value equals to zero, then there is no agreement between raters, if the kappa value is equal to 1, then there is full agreement. This interpretation will help decide if the results from the testing phase agree in any way, the higher the agreement the more likely the user will benefit from the algorithm they are measured against.

In the testing phase I will need the input of users to measure against algorithms, if the choices in senses are to a degree the same, then the highest scoring algorithm should act as the background WSD method.

Fleiss Kappa

This kappa measure does the same as Cohens Kappa, the difference is that it can measure agreement between more than two raters. Meaning that it takes into consideration the rate of chance agreement. In this case, there are fourteen different raters rating the same items.

8 RESULTS

This subsection presents the results from the last major comparison between the algorithms and human tests. First, the algorithms and human test results will be presented in the same order as presented in the development section. A comparison of the different disambiguation algorithms follows after the general comparison.

8.1 Survey Results

The questionnaire has revealed some clarity to the issues revealed from the previous questionnaire. The results range from being in total agreement to total disagreement. Total disagreement means that the word meanings are spread out over both senses and part of speech elements. One example of total agreement is the question regarding the word “tanks” in the survey (see appendix B), has the context: "Germany's combination of fast armoured tanks on land, and superiority in the air, made a unified attacking force".

The word in this context is obviously concerned with military armoured vehicles, the list of meanings from WordNet is numbered at eight different senses. In this case the senses are clearly distinguishable from one another and the sentence is unambiguous, the number of senses are sparse compared to others as well. In other words, the circumstances for correctly deciding the correct sense are optimal. Google cake diagram presented below:

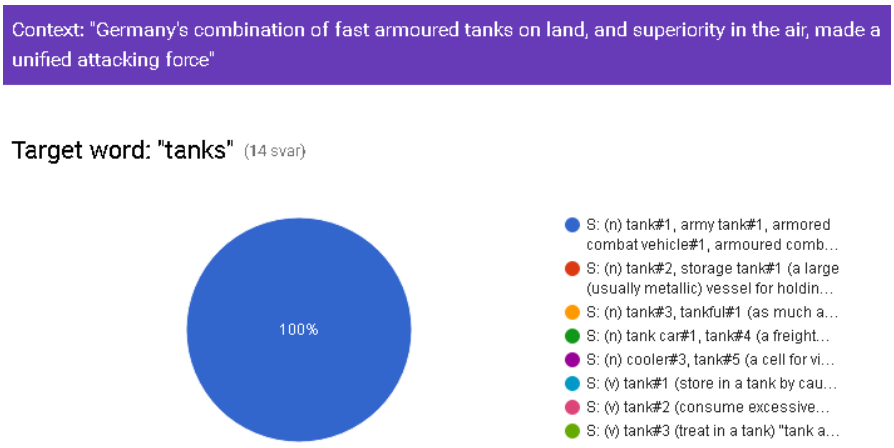

Figure 8-1: Google form questionnaire sample

The counter opposite is the question with the target word building “building”. In WordNet “building” has fourteen meanings divided over verbs and nouns, the answers from the survey spread over seven different meanings, four verbs and three nouns. The top two accumulated to 42,8 percent of the answers, both verbs, meanings and the google form sample is presented below.

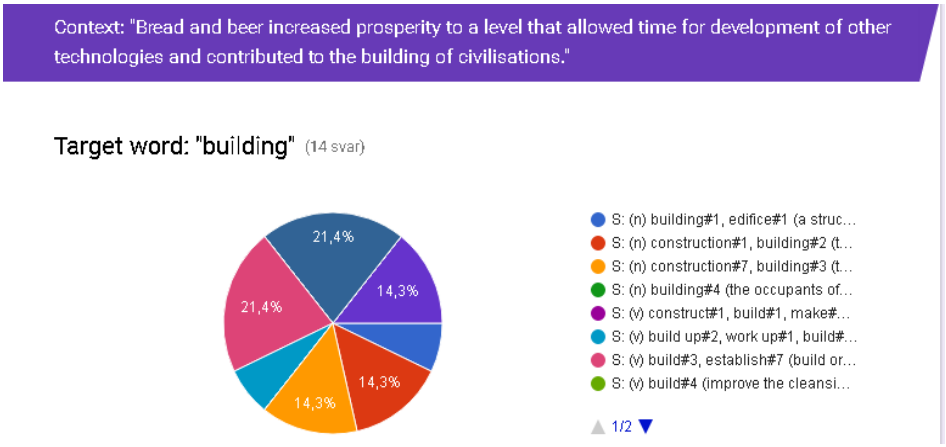

Figure 8-2: Google form questionnaire sample 2

The senses of the top two in figure 11-2, are below:

S: (v) build#6 (give form to, according to a plan) ***"build a modern nation"; "build a million-dollar business"***

And

S: (v) build#3, establish#7 (build or establish something abstract) ***"build a reputation"***

The disagreement within this question is severe in comparison to the previous example, every answer divided over seven different meanings, and the top two answers have different definitions but the distinction between the two of them is vague. The senses are related in the fact that they both build abstract things, i.e. a sample of fine grained polysemy.

Rater agreement

From the two examples above one can see agreement ranging from full agreement to almost none. Running an agreement measure on every raters individual choice for every question in the experiment sheds some light on overall agreement. Fleiss's Kappa (Fleiss 1971) agreement test on the users decisions are presented below:

Fleiss' Kappa for m Raters

Raters/Subjects	14
Kappa	0.407

Table 8-1:Fleiss Kappa on participants

Trouble

Conversing with some of the subjects who subjected themselves to the questionnaire have all issued the same concern, especially in the ways of the length and time consumption. Apparently, two even quit the survey while they were still carrying it out due to its extensive nature.

8.2 Algorithm results

The results from the algorithms are presented in this subsection.

The numbers from each of the algorithms vary, the pre-existing algorithms are not able to disambiguate other than nouns, nouns and verbs, verbs. That excludes some of the lines from the testing phase (see Appendix F), since the target word in a number of lines are adjectives, or possibly adverbs. Below is an error count:

Algorithm error count

Algorithms	**Missed**
WordToSet	2
AllWords	18
TargetWord	15
NLTK	16
Manual1	8
Manual2	0
Manual3	0
Manual4	27

Table 8-2: Algorithm error count (100 sentence-target test)

Kappa Measurements

Running Fleiss Kappa test on the algorithms, would not produce relevant results for this thesis, instead, running Cohens Kappa on the top human answers and top results from the algorithms would produce an answer to which degree the algorithms are in agreement with humans.

Presented below are the individual scores for the pre-existing algorithms and their measurement versus the top human answer:

Top Algorithm-Top with Human Cohen's Kappa

Algorithms	**Subjects**	**Kappa**
TargetWord	85	0.0465
WordToSet	98	0.22
AllWords	82	0.242
NLTK	83	0.143
Manual1	92	0.205
Manual2	100	0.208
Manual3	100	0.218
Manual4	73	0.162

Table 8-3: Algorithms Vs Top human

The following measurements are almost the same as above, the difference is that now, the measurements are done with the top algorithm disambiguation versus the top **two** user disambiguation. A subjective evaluation of the human responses suggested that in the majority of cases at least the top two choices are examples of fine grained polysemy.

The table presented below shows the agreement between algorithm and the adjusted human response:

Top Algorithm-Top Two Human Cohen's Kappa

Algorithms	Subjects	Kappa
TargetWord	85	0.151
WordToSet	98	0.371
AllWords	82	0.424
NLTK	83	0.251
Manual1	92	0.324
Manual2	100	0.358
Manual3	100	0.378
Manual4	73	0.258

Table 8-4: Algorithms Vs Top two human

As a final test we evaluate the usefulness of the disambiguation algorithms in the semantic tagging task. This would be the case if we could order the disambiguation senses and one of the top three coincided with the correct disambiguated sense. The table below presents the numbers from where I have retrieved the top three senses from the algorithms and top two human results (as in the previous table). I have left the top guess if it is the same as the human choice, otherwise inserted the second guess if it agrees, otherwise inserted the third guess. If none of the top three agree, leave it with the first guess. Notice that the only commonly available library which can return a list of disambiguations is the WordToSet algorithm.

Results presented below:

Top Three Algorithms – Top Two human Cohen's Kappa

Algorithms	Subjects	Kappa
WordToSet	98	0.681
Manual1	92	0.486
Manual2	100	0.668
Manual3	100	0.708
Manual4	73	0.545

Table 8-5: Top three Algorithm Vs Top two human

9 DISCUSSION

This section discusses the findings presented in the Result section.

9.1 Kappa measurement between Algorithms and Top Human choice

Measuring strictly to the top human choice and the top algorithm choice, reveals that the AllWords technique has the highest agreement with humans with a Kappa score of 0,242 which is interpreted as a fair agreement (see Table 7.1). This is the second lowest interpretation if one omits less than zero agreement ("poor" agreement). The AllWords algorithm is the latest of the SenseRelate Perl modules(Pedersen and Kolhatkar 2009), which could suggest that the disambiguation methods have been improved upon over time. The measurements do not mean that the others are incompetent in comparison, it means that the AllWords algorithm agrees more with this human results in this survey than the others when it comes to top results, and this includes the manual algorithms. The closest of the manual algorithms was Manual3 algorithm, which achieved a 0.218 agreement score. According to the interpretation it is the same as the AllWords method, however with lower agreement score.

The manual algorithms all have a 'fair' or less agreement score from the Kappa testing.

In all, the average from both pre-existing and the thesis created methods differ in only 1 percent. In average the numbers have not such a big difference, but individually, it is clear that the agreements lie closer with the manual methods than with the pre-existing algorithms.

The Kappa score is pretty low overall when measured strictly against the top disambiguation, measuring the top choice from the algorithm against the top two survey choice, the agreement changes drastically. TargetWord which is the algorithm who receives the lowest agreement score overall, elevate times 3.2 score (Kappa = 0,151). Indicating that in the case of TargetWord, that more of the agreed upon senses lie in the second choice. The AllWords method still has the greatest agreement score (Kappa = 0,424 = "moderate") of all the algorithms. Meaning that, adding the second top choice to the measurement clearly includes a big difference in Kappa measurements.

9.2 Algorithms created for the task

The agreement lies closer to human choice in the pre-existing algorithms, which suggests that if an algorithm were to be used for defining a words meaning for people, it should be the AllWords method. However, the word meaning would be what the user wants about 42 percent of the time. The manual algorithms are designed with the thesis goal in mind, this means that the algorithms return a ranked list of top choices. Based on that, the manual algorithms top three choices measured against the top two choices of humans should return a greater score of agreement.

From table 8-4, one can see differences. The first difference is that the pre-existing methods are excluded, the reason for this is that these methods do not include a function to retrieve a list of the top senses. The second observation is that the kappa agreement scores are noticeably greater than in table 8-3. This means that in the top three meanings returned from the algorithms there were senses that agreed with either of the top two from the humans. Manual3 was the one with the greatest kappa agreement, which suggests that manual3 returns the list most likely to agree with what humans would choose. The kappa measure was 0.708, and is interpreted as a significant agreement. That manual3 would be the one with the greatest agreement was suspected, since the algorithm takes into account the heading when disambiguating against the context. Not expected, is that the WordToSet' agreement score was the second highest of the algorithms with the ability to return a list of ranked senses. With a kappa agreement score of 0,681, a substantial agreement.

The reasoning behind this measurement is to hopefully eliminate the fine granularity of the top sense choices in both human and algorithm. The theory is that the top human choices are closely related but have different meanings like polysemous words, that these are the senses that is difficult to differentiate between. The kappa score increasing as much as it is shown in table 6, suggests that the reasoning is correct, at least in agreement with humans.

Discussed in the previous subsection is the fact that sense one, two, and three of a respected POS element appear 74 percent of the time in the survey. Meaning that 1035 out of 1400 answers was one of the first three senses in WordNet. This suggests the theory that the text such as the ones I have procured for the survey and algorithm testing are of a natural kind. Unspecific enough to avoid the specialized senses that some words have. Compared to the algorithms, Manual3 is the one who comes closest to this with an agreement of 0,708 kappa score. Manual3 has chosen the one of the three first senses 76 percent of the time. This could indicate that the use of WSD in web pages containing general language, could be redundant.

An overall observation is that agreement was quite low across the board. For example, in Senseval-3(Agirre and Edmonds 2006) for English disambiguation algorithms was getting about 65 percent correct senses. Which is very high in comparison to the table 8.3 where the top agreement score was 0.242, this raises the possibility that our human raters were somehow influencing the results in a negative way.

9.3 Human anomaly

To follow up on the possibility that human raters were confused by the fine grained polysemy in WordNet, we investigated the diversity of choices they made. The table below shows that 73 percent of the choices were from the top three senses, even though the materials were designed such that the suggested best sense was more broadly distributed amongst the possible senses.

# senses	accumulated	percentage
1	470	33,6
1,2	830	59,3
1,2,3	1035	73,9

Table 9-1: first, second, and third sense count and percentage

Table 9-6 presents numbers that could prove why the algorithms have such a low kappa score when measured against humans. 73 percent of the answers are within the first three in every POS element available could indicate several things. One aspect of this could mean that when humans are reviewing the answers and discovering a "close enough" sense earlier in the answer collection. Implying that a participants' incentive to read closer in the following senses, drops if it is good enough. This could indicate that the first senses are more general, and covers more with its definition than the senses located later in the collection. Possibly skipping the more specific sense that would fit more in that context. From this point, the algorithm could have chosen the more sensible sense, and still would have gotten a wrong answer due to human "laziness". Meaning that all of the algorithms could have gotten a much higher agreement score than recorded in section 8. A possible solution to the fine grained problem could be to train more people better to distinguish between the difficult senses. Which could give a better kappa score between humans. Maybe have expert participants join the group. This could produce a bell curve where the most sensible sense is positioned near or at the peak.

Another aspect could be the granularity mentioned through the thesis. The fact that the fine granularity WordNet has between senses are too closely defined. Making it difficult for people to tell apart senses. Two examples are of the words: "career", and "light". Both are positioned at each end of the spectre when it comes to numbers of senses, both have so closely defined senses that it is impossible to see the difference. Career has two noun senses and are defined as so:

Career:

1. the particular occupation for which you are trained.
2. the general progression of your working or professional life.

Break has 59 different verb senses, and listing them all would not ideal. Instead here are two sense definitions that basically means the same thing.

1. destroy the integrity of; usually by force; cause to separate into pieces or fragments.
2. render inoperable or ineffective.

Another example of such a case is the word "split", which through WordNet has eighteen different definitions divided over nouns, verbs, and adjectives. Examining Appendix B, one can see from the question which involves "split" as the target word, that within that particular context the word is a verb, and that the answers options for verbs are not very relatable to the context. However, both the first and the third synset has been chosen three times. The users picked eight different senses, which suggests a difficulty in finding the difference between different meanings. The algorithms however, have all decided that "split" is a noun, and measured thereafter.

Another sample is from the first question, it involves disambiguating "hen" from the context: "A male duck is called a drake and the female duck is called a duck, or in ornithology a "***hen***". In this case one can argue that the correct meaning of the word should be the second sense of the word. The algorithms disagree on which sense it should be defined as. Presented below are the top result from each algorithm regarding that particular disambiguation.

AllWords WSD	Word to Set WSD	Target Word WSD	NLTK WSD	Manual WSD I (Python)	Manual WSD II (Perl)	Manual WSD III (Perl)	Manual WSD IV (Perl)
hen#n#3	hen#n#1	hen#n#4	hen#n#4	hen#n#2	hen#n#1	hen#n#3	hen#n#2

Table7: Algorithm test sample

The senses are defined as so:

- (noun) **hen**, biddy (adult female chicken)
- (noun) **hen** (adult female bird)
- (noun) **hen** (flesh of an older chicken suitable for stewing)
- (noun) **hen** (female of certain aquatic animals e.g. octopus or lobster)

From that question the correct sense should be hen#n#2, and is defined as an adult female bird. The answers from the questionnaire are divided equally between the first and second sense.

One of the cases where the contrast between senses have close to no contrast is the question with the target word: "**domestic"** (See Appendix B). The top two senses chosen by humans are presented below:

- **domestic#4**, domesticated#1 (converted or adapted to domestic use) "domestic animals"; "domesticated plants like maize"
- **domestic#2** (of or relating to the home) "domestic servant"; "domestic science"

The two having lightly contrasted differences, it is easy to see the difficulty in distinguishing them. Cases like this, could support the need for a revised WordNet, where word meanings with fine differences could be collapsed. Doing a subjective research between top two answers reveal that 95 percent are fine grained senses. The participants in this survey have English as their second language, this could be a contributing factor to why most of the answers are divided into two or more senses, that the distinction between part of speech elements seemingly is difficult, and why the agreement score are so low.

10 CONCLUSIONS

This section will conclude the thesis by answering the thesis questions.

10.1 Thesis Questions

Q1: Can existing WSD algorithms accurately predict the correct sense of a target word in a webpage?

The pre-existing algorithms were not as capable when measured against what humans chose as the correct answer. The algorithm who did the best in the base case was AllWords with an agreement score of 0.242. However, the AllWords algorithm also missed the most of the disambiguations by a number of 18 missed out of a hundred. Against the top two, AllWords had again the top agreement score with a 0.424 kappa agreement. So, the agreement interpreted from the kappa score was a high as "fair", and is lower than we initially expected. However, note from the discussion that this might be in part because of the human responses.

Q2. Can we construct more accurate algorithms by considering the standard HTML elements like titles as a contextual element?

Yes, the manual3 is able to get more accurate senses with the help of HTML headings. However, it is second behind the AllWords algorithm. AllWords gave the top agreement score for the top human answers.

Q3. Can we use the disambiguating algorithms to assist users in tasks like semantic tagging, or other forms of manual markup?

According to the agreement score, which returns a significant agreement between humans and manual3 method. This proves that when a user has this program running in the background of an e.g. semantic tagging program, and wants to find the sense of a word in a HTML document. Manual3 algorithm would present a list of probable senses that would be usable for the person 71 percent of the time. This would cut down the tagging time when manually looking for the correct sense of a word to tag with data.

References

ACL, Wiki. 2013. "POS Tagging (State of the Art)." http://aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art).

Agirre, Eneko, and Philip Edmonds. 2006. "Word Sense Disambiguation: Algorithms and Applications." *Text Speech and Language Technology* 33 (33): 384. http://books.google.com/books?hl=en&lr=&id=GLck75U20pAC&oi=fnd&pg=PR13&dq=Word+Sense+Disambiguation:+Algorithms+and+Applications&ots=M5pyjmIOv7&sig=Z7y-tiU-_sRFmEQYPHK4fU4y8fY\nhttp://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/140.

Agirre, Eneko, Oier Lopez De Lacalle, and Aitor Soroa. 2009. "Knowledge-Based Wsd on Specific Domains: Performing Better than Generic Supervised WSD." *IJCAI International Joint Conference on Artificial Intelligence*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.150.682&rep=rep1&type=pdf.

Banerjee, Satanjeev, and Ted Pedersen. 2002. "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet." *Computational Linguistics and Intelligent Text ...*. http://link.springer.com/chapter/10.1007/3-540-45715-1_11\nhttps://www.d.umn.edu/~tpederse/Pubs/banerjee.pdf.

Brill, Eric. 1992. "A Simple Rule-Based Part of Speech Tagger." *Applied Natural Language*, 3. doi:10.3115/1075527.1075553.

Cohen, Jacob. 1960. "A Coefficient of Agreement for Nominal Scales." *Educational and Psychological Measurement* 20: 37–46. doi:10.1177/001316446002000104.

Dash, Niladri Sekhar. 2001. "Polysemy and Homonymy : A Conceptual Labyrinth."

Fellbaum, Christiane. 2000. "Book Reviews." *Contemporary Physics*. doi:10.1080/001075100750827151.

Fleiss, Joseph L. 1971. "Measuring Nominal Scale Agreement among Many Raters." *Psychological Bulletin*. doi:10.1037/h0031619.

G, U T N N. 2016. "Quantitative Studies - How Many Users to Test.pdf." *June 6*. https://www.nngroup.com/articles/quantitative-studies-how-many-users/.

Hearst, M. 1999. "Untangling Text Data Mining." *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 3–10. doi:10.3115/1034678.1034679.

Hirst, Graeme, and David St-Onge. 1998. "Lexical Chains as Representations of Context for

the Detection and Correction of Malapropisms." *WordNet - An Electronic Lexical Database*, no. April: 305–32. doi:citeulike-article-id:4893262.

Kanimozhi, K.V., and Dr.M. Venkatesan. 2015. "Unstructured Data Analysis-A Survey." *Ijarcce* 4 (3): 223–25. doi:10.17148/IJARCCE.2015.4354.

Landis, J R, and G G Koch. 1977. "The Measurement of Observer Agreement for Categorical Data." *Biometrics* 33 (1): 159–74. doi:10.2307/2529310.

Leacock, Claudia, G a Miller, and Martin Chodorow. 1998. "Using Corpus Statistics and WordNet Relations for Sense Identification." *Computational Linguistics* 24 (1): 147–65. doi:10.3115/1075812.1075867.

Lesk, Michael. 1986. "Automatic Sense Disambiguation Using Machine Readable Dictionaries." In *Proceedings of the 5th Annual International Conference on Systems Documentation - SIGDOC '86*, 24–26. doi:10.1145/318723.318728.

Lin, Dekang. 1998. "An Information-Theoretic Definition of Similarity." *Proceedings of ICML*, 296–304. doi:10.1.1.55.1832.

Manning, Christopher D. 2011. "Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?" *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6608 LNCS (PART 1): 171–89. doi:10.1007/978-3-642-19400-9_14.

McCarthy, Diana. 2006. "Relating WordNet Senses for Word Sense Disambiguation." *Proceedings of the ACL Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, 17–24.

NAVARRE, S, and H STEIMAN. 2002. *Root-End Fracture During Retropreparation: A Comparison Between Zirconium Nitride-Coated and Stainless Steel Microsurgical Ultrasonic Instruments*. *Journal of Endodontics*. Vol. 28. doi:10.1097/00004770-200204000-00018.

NLTK. 2015. "Natural Language Toolkit." *Nltk*. http://www.nltk.org/.

Pedersen, Ted, and Varada Kolhatkar. 2009. "WordNet :: SenseRelate :: AllWords - A Broad Coverage Word Sense Tagger That Maximizes Semantic Relatedness." *Proceeding NAACL-Demonstrations '09 Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session*, 17–20. http://dl.acm.org/citation.cfm?id=1620964.

Pesaranghader, Ahmad, Ali Pesaranghader, and Norwati Mustapha. 2014. "Word Sense

Disambiguation for Biomedical Text Mining Using Definition-Based Semantic Relatedness and Similarity Measures." *International Journal of Bioscience, Biochemistry and Bioinformatics* 4 (4): 280–83. doi:10.7763/IJBBB.2014.V4.356.

Resnik, Philip. 1995. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy." *Roceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1 - IJCAI'95* 1: 6. doi:10.1.1.55.5277.

Specia, Lucia, Mark Stevenson, Regent Court, Portobello Street, Gabriela Castelo, and Branco Ribeiro. 2005. "Multilingual versus Monolingual WSD." *Language*, 33–40.

Vickrey, David, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. "Word-Sense Disambiguation for Machine Translation." *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05)*, 771–78. doi:10.3115/1220575.1220672.

"WordNet Stats 2016-11-23." 2016. Accessed November 23. http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html#sect2.

Wu, Zhibiao, and Martha Palmer. 1994b. "Verb Semantics and Lexical Selection." *32nd Annual Meeting on Association for Computational Linguistics*, 133–38. doi:10.3115/981732.981751.

———. 1994a. "Verb Semantics and Lexical Selection." In *32nd Annual Meeting on Association for Computational Linguistics*, 133–38. Morristown, NJ, USA: Association for Computational Linguistics. doi:10.3115/981732.981751.

Zhong, Zhi, and Hwee Tou Ng. 2012. "Word Sense Disambiguation Improves Information Retrieval." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, no. July: 273–82. http://www.aclweb.org/anthology/P12-1029.

Appendix A – Testing Database w/ HeadLines

https://en.wikipedia.org/wiki/Duck|hen|Duck|A male duck is called a drake and the female duck is called a duck, or in ornithology a hen

https://en.wikipedia.org/wiki/Duck|fly|Duck|To be able to submerge more easily, the diving ducks are heavier than dabbling ducks, and therefore have more difficulty taking off to fly.

https://en.wikipedia.org/wiki/Duck|breeding|Duck|Ducks also tend to make a nest before breeding, and after hatching to lead their ducklings to water.

https://en.wikipedia.org/wiki/Duck|quack|Duck|A common urban legend claims that duck quacks do not echo

https://en.wikipedia.org/wiki/Duck|domestic|Duck|Almost all the varieties of domestic ducks are descended from the mallard (Anas platyrhynchos)

http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|vicious|5 Ways Pirates Were Way More Modern Than You Realize|We tend to think of pirates as bloodthirsty thieves, brutal rapists, and vicious murderers,

http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|shooting|5 Ways Pirates Were Way More Modern Than You Realize|Beyond that, when a typical workday consists of shooting terrified sailors in the face while rival pirates hurl themselves at you with blood-tarnished daggers clenched between their teeth,

http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|defeat|5 Ways Pirates Were Way More Modern Than You Realize|So, alongside an English pirate captain named John Ward, Danziger showed the Algerians both how to build European style ships and how to defeat them in combat, allowing Algiers to extend its piratical reach all over Europe.

http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|career|5 Ways Pirates Were Way More Modern Than You Realize|piracy was an attractively viable career choice.

http://www.cracked.com/article_20448_5-ways-pirates-were-way-more-modern-than-you-realize.html|royal|5 Ways Pirates Were Way More Modern Than You Realize|Actually, Jack Ward, a wildly successful pirate captain who deserted from the Royal Navy along with 30 of his shipmates (because high-seas robbery is way more lucrative than being in the military)

https://en.wikipedia.org/wiki/Snake|evolved|Snake|Snakes are thought to have evolved from either burrowing or aquatic lizards, perhaps during the Jurassic period

https://en.wikipedia.org/wiki/Snake|scales|Snake|Most snakes use specialized belly scales to travel, gripping surfaces.

https://en.wikipedia.org/wiki/Snake|ribs|Snake|The skeleton of most snakes consists solely of the skull, hyoid, vertebral column, and ribs

https://en.wikipedia.org/wiki/Snake|fangs|Snake|A poison is inhaled or ingested, whereas venom produced by snakes is injected into its victim via fangs

https://en.wikipedia.org/wiki/Snake|structure|Snake|The snake's jaw is a complex structure

http://www.bbc.co.uk/history/worldwars/wwtwo/ww2_summary_01.shtml|fought|World War Two: Summary Outline of Key Events|Denmark surrendered immediately, but the Norwegians fought on

http://www.bbc.co.uk/history/worldwars/wwtwo/ww2_summary_01.shtml|tanks|World War Two: Summary Outline of Key Events|Germany's combination of fast armoured tanks on land, and superiority in the air, made a unified attacking force

http://www.bbc.co.uk/history/worldwars/wwtwo/ww2_summary_01.shtml|control|World War Two: Summary Outline of Key Events|With continental Europe under Nazi control, and Britain safe

http://www.bbc.co.uk/history/worldwars/wwtwo/ww2_summary_01.shtml|surprise|World War Two: Summary Outline of Key Events|The Japanese, tired of American trade embargoes, mounted a surprise attack on the US Navy base of Pearl Harbor

http://www.bbc.co.uk/history/worldwars/wwtwo/ww2_summary_01.shtml|sickening|World War Two: Summary Outline of Key Events|The New Year saw the Soviet liberation of Auschwitz, and the revelation of the sickening obscenity of the Holocaust

http://www.imdb.com/title/tt1638355/|mission|The Man from U.N.C.L.E. (2015)|In the early 1960s, CIA agent Napoleon Solo and KGB operative Illya Kuryakin participate in a joint mission against a mysterious criminal organization, which is working to proliferate nuclear weapons.

http://www.imdb.com/title/tt1638355/|match|The Man from U.N.C.L.E. (2015)|As they make their way to the border they're chased by Kuryakin, who turns out to be Solo's match.

http://www.imdb.com/title/tt1638355/|brains|The Man from U.N.C.L.E. (2015)|The business has been in Alexander's (Luca Calvani) family but actually his wife Victoria (Elizabeth Debicki) is the brains behind the operation.

http://www.imdb.com/title/tt1638355/|crashes|The Man from U.N.C.L.E. (2015)|Solo crashes the party and gets in a fight with security because he doesn't present an invitation.

http://www.imdb.com/title/tt1638355/|short|The Man from U.N.C.L.E. (2015)|Fortunately, there's a short in the electric chair

https://en.wikipedia.org/wiki/Boxing|game|Boxing|While people have fought in hand-to-hand combat since before the dawn of history, the origin of boxing as an organized sport may be its acceptance by the ancient Greeks as an Olympic game in BC 688.

https://en.wikipedia.org/wiki/Boxing|deaths|Boxing|The first boxing rules, called the Broughton's rules, were introduced by champion Jack Broughton in 1743 to protect fighters in the ring where deaths sometimes occurred.

https://en.wikipedia.org/wiki/Boxing|damage|Boxing|Amateur boxing has a point scoring system that measures the number of clean blows landed rather than physical damage

https://en.wikipedia.org/wiki/Boxing|quit|Boxing|Through the early twentieth century, it was common for fights to have unlimited rounds, ending only when one fighter quit

https://en.wikipedia.org/wiki/Boxing|power|Boxing|A brawler is a fighter who generally lacks finesse and footwork in the ring, but makes up for it through sheer punching power.

https://www.olympic.org/weightlifting-equipment-and-history|measure|Weightlifting Equipment and History - Olympic Sport History|As a means to measure strength and power, weightlifting was practised both by ancient Egyptian and Greek societies

https://www.olympic.org/weightlifting-equipment-and-history|games|Weightlifting Equipment and History - Olympic Sport History|From the 2000 Olympic Games in Sydney, men have competed in eight weight categories and women in seven

https://www.olympic.org/weightlifting-equipment-and-history|origins|Weightlifting Equipment and History - Olympic Sport History|Weightlifting has ancient origins. It featured at the first modern Olympic Games in Athens in 1896.

https://www.olympic.org/weightlifting-equipment-and-history|gold|Weightlifting Equipment and History - Olympic Sport History|Turkey’s and Halil Mutlu have each won three gold medals, like Greece’s Pyrros Dimas and Kakhi Kakhiasvilis

https://www.olympic.org/weightlifting-equipment-and-history|programme|Weightlifting Equipment and History - Olympic Sport History|Although men’s weightlifting has always been on the programme of the Olympic Games- except for at the 1900, 1908 and 1912 editions – women started to participate only at the 2000 Games in Sydney.

http://www.touropia.com/famous-underground-caves-in-the-world/|underground|10 Famous Underground Caves in the World|For the less adventurous, a number of the most beautiful

underground caves have been converted into show caves, where artificial lighting, floors, and other aids allow the casual tourist to experience the cave with minimal inconvenience.

http://www.touropia.com/famous-underground-caves-in-the-world/|froze|10 Famous Underground Caves in the World|The ice formations in the cave were formed by thawing snow which drained into the cave and froze during winter.

http://www.touropia.com/famous-underground-caves-in-the-world/|mouth|10 Famous Underground Caves in the World|According to a legend, Reed Flute Cave got its name because people believed that the reed by the cave's mouth could be made into flutes

http://www.touropia.com/famous-underground-caves-in-the-world/|grow|10 Famous Underground Caves in the World|The crystals became so large because of the extremely hot temperatures inside the cave, reaching a steamy 58 degrees Celsius (136 degrees Fahrenheit), that allowed microscopic crystals to form and grow.

http://www.touropia.com/famous-underground-caves-in-the-world/|smooth|10 Famous Underground Caves in the World|The lower gallery which has an overall length of 6,200 meters (20,300 feet) is located 60 meters (200 feet) below the upper gallery. It is traversed by a smooth underwater river and a lake.

http://www.history.com/news/history-lists/8-things-you-should-know-about-al-capone|fight|8 Things You Should Know About Al Capone|In 1917, Capone's face was slashed during a fight at the Harvard Inn, after he insulted a female patron and her brother retaliated, leaving him with three indelible scars.

http://www.history.com/news/history-lists/8-things-you-should-know-about-al-capone|served|8 Things You Should Know About Al Capone|Capone would attempt to shield the scarred side of his face in photographs, and tried to write them off as war wounds—although he never served in the military.

http://www.history.com/news/history-lists/8-things-you-should-know-about-al-capone|offense|8 Things You Should Know About Al Capone|Ninety percent of the people of Cook County drink and gamble and my offense has been to furnish them with those amusements.

http://www.history.com/news/history-lists/8-things-you-should-know-about-al-capone|stunned|8 Things You Should Know About Al Capone|The crime became known as the St. Valentine's Day Massacre and stunned the nation.

http://www.history.com/news/history-lists/8-things-you-should-know-about-al-capone|charges|8 Things You Should Know About Al Capone|in October 1931, the all-male

jury (Illinois didn't allow female jurors until 1939) found the gangster guilty of five charges (three felonies and two misdemeanors) of the more than 20 counts against him

https://authoritynutrition.com/top-13-evidence-based-health-benefits-of-coffee/|healthy|13 Health Benefits of Coffee, Based on Science|Coffee is actually very healthy It is loaded with antioxidants and beneficial nutrients that can improve your health.

https://authoritynutrition.com/top-13-evidence-based-health-benefits-of-coffee/|natural|13 Health Benefits of Coffee, Based on Science|There's a good reason for that… caffeine is one of the very few natural substances that have actually been proven to aid fat burning.

https://authoritynutrition.com/top-13-evidence-based-health-benefits-of-coffee/|flight|13 Health Benefits of Coffee, Based on Science|This is the fight or flight hormone, designed to make our bodies ready for intense physical exertion.

https://authoritynutrition.com/top-13-evidence-based-health-benefits-of-coffee/|elevated|13 Health Benefits of Coffee, Based on Science|It is characterized by elevated blood sugars in the context of insulin resistance or an inability to secrete insulin.

https://authoritynutrition.com/top-13-evidence-based-health-benefits-of-coffee/|risk|13 Health Benefits of Coffee, Based on Science|Some studies also show that coffee drinkers have a 20% lower risk of stroke

http://www.badassoftheweek.com/index.cgi?id=303556515876|true|Harald Fairhair|Harald Fairhair was the first true King of Norway, the national hero of his country,

http://www.badassoftheweek.com/index.cgi?id=303556515876|machine|Harald Fairhair|a mega-tough human killing machine known as Duke Guthorm.

http://www.badassoftheweek.com/index.cgi?id=303556515876|sneak|Harald Fairhair|Then, when he heard some other guy was trying to overthrow Harald, the young king sent Viking warriors to sneak into that dude's city and set fire to his castle in the middle of the night

http://www.badassoftheweek.com/index.cgi?id=303556515876|escape|Harald Fairhair|When the wannabe future king and his men ran outside to escape the blaze, they ran right into the spears of Harald Fairhair's warriors.

http://www.badassoftheweek.com/index.cgi?id=303556515876|fair|Harald Fairhair|Unstoppable, strong-willed, and energetic, King Harald was also fair.

https://en.wikipedia.org/wiki/Beer|hops|Beer|Most beer is flavoured with hops, which add bitterness and act as a natural preservative, though other flavourings such as herbs or fruit may occasionally be included.

https://en.wikipedia.org/wiki/Beer|building|Beer|Bread and beer increased prosperity to a level that allowed time for development of other technologies and contributed to the building of civilisations.

https://en.wikipedia.org/wiki/Beer|quality|Beer|In 1516, William IV, Duke of Bavaria, adopted the Reinheitsgebot (purity law), perhaps the oldest food-quality regulation still in use in the 21st century,

https://en.wikipedia.org/wiki/Beer|control|Beer|The development of hydrometers and thermometers changed brewing by allowing the brewer more control of the process and greater knowledge of the results.

https://en.wikipedia.org/wiki/Beer|pale|Beer|Pale ale is a beer which uses a top-fermenting yeast[95] and predominantly pale malt. It is one of the world's major beer styles.

https://en.wikipedia.org/wiki/Nuclear_weapon|range|Nuclear weapon|The amount of energy released by fission bombs can range from the equivalent of just under a ton to upwards of 500,000 tons

https://en.wikipedia.org/wiki/Nuclear_weapon|split|Nuclear weapon|All fission reactions necessarily generate fission products, the radioactive remains of the atomic nuclei split by the fission reactions

https://en.wikipedia.org/wiki/Nuclear_weapon|tests|Nuclear weapon|Only six countries—United States, Russia, United Kingdom, People's Republic of China, France and India—have conducted thermonuclear weapon tests

https://en.wikipedia.org/wiki/Nuclear_weapon|core|Nuclear weapon|There are two types of boosted fission bomb: internally boosted, in which a deuterium-tritium mixture is injected into the bomb core, and externally boosted, in which concentric shells of lithium-deuteride and depleted uranium are layered on the outside of the fission bomb core.

https://en.wikipedia.org/wiki/Nuclear_weapon|tapping|Nuclear weapon|The concept involves the tapping of the energy of an exploding nuclear bomb to power a single-shot laser which is directed at a distant target.

http://www.timephysics.com/|wrapped|WHAT IS TIME AND WHAT CAUSES TIME?|We measure time, keep time, meet and greet in time and our daily lives are completely wrapped around the onward rush of time.

http://www.timephysics.com/|picture|WHAT IS TIME AND WHAT CAUSES TIME?|Events in time are always accompanied with a mental picture of a place suggesting that time is a dimension

http://www.timephysics.com/|radial|WHAT IS TIME AND WHAT CAUSES TIME?|The use of units like seconds and minutes which are radial angle measurements in geometry may be pointing toward the original connection of time measurements to radial motion of astronomical objects across the sky.

http://www.timephysics.com/|motion|WHAT IS TIME AND WHAT CAUSES TIME?|As this thought experiment also can be extended to particles held together by electromagnetic forces we can say that time involves both motion and forces.

http://www.timephysics.com/|block|WHAT IS TIME AND WHAT CAUSES TIME?|In the block universe time is laid out as a time-scape similar to landscape and it is obvious that there cannot be a free will.

http://starwars.wikia.com/wiki/Anakin_Skywalker/Legends|served|Anakin Skywalker|Anakin Skywalker was a Force-sensitive Human male who served the Galactic Republic as a Jedi Knight and later served the Galactic Empire as the Sith Lord Darth Vader.

http://starwars.wikia.com/wiki/Anakin_Skywalker/Legends|piloting|Anakin Skywalker|At an early age, Skywalker exhibited signs of Force-sensitivity; he could sense things before they happened, occasionally sensed the emotions of others, and had exceptional piloting skills which Watto put to use when he made the child race his podracer

http://starwars.wikia.com/wiki/Anakin_Skywalker/Legends|build|Anakin Skywalker|Skywalker began to build both a podracer from the pieces he could salvage from the junkyards

http://starwars.wikia.com/wiki/Anakin_Skywalker/Legends|fire|Anakin Skywalker|However, the Dug Khiss entered the warehouse at that point and opened fire on the two boys

http://starwars.wikia.com/wiki/Anakin_Skywalker/Legends|manned|Anakin Skywalker|Skywalker manned the shop while Watto took the man, Qui-Gon Jinn, into the junkyard.

http://www.bbc.co.uk/scotland/history/articles/william_wallace/|fine|william wallace|This is the truth I tell you: of all things freedom's most fine. Never submit to live, my son, in the bonds of slavery entwined.

http://www.bbc.co.uk/scotland/history/articles/william_wallace/|base|william wallace|From his base in the Ettrick Forest his followers struck at Scone, Ancrum and Dundee

http://www.bbc.co.uk/scotland/history/articles/william_wallace/|struck|william wallace|His MacDougall allies cleared the west, whilst he struck through the north east.

http://www.bbc.co.uk/scotland/history/articles/william_wallace/|ladder|william wallace|Wallace's extraordinary military success catapulted him to the top of the social ladder.

http://www.bbc.co.uk/scotland/history/articles/william_wallace/|submit|william wallace|With no prospect of victory, the Scottish leaders capitulated and recognised Edward as overlord in 1304. Only Wallace refused to submit, perhaps signing his own death warrant at this time.

http://www.telegraph.co.uk/film/terminator-genisys/timeline-franchise/|simple|Terminator timeline: the (extremely confusing) story so far|In the first Terminator movie, things are simple enough. By "simple", we mean mildly confusing and paradoxical, because that's always the way with time travel

http://www.telegraph.co.uk/film/terminator-genisys/timeline-franchise/|race|Terminator timeline: the (extremely confusing) story so far|in 2029, the Future War, between the human race and machines (led by an AI system called Skynet) is ongoing.

http://www.telegraph.co.uk/film/terminator-genisys/timeline-franchise/|young|Terminator timeline: the (extremely confusing) story so far|A Terminator (Arnold Schwarzenegger) is consequently sent back to 1984 to kill John's mother, a young waitress named Sarah Connor

http://www.telegraph.co.uk/film/terminator-genisys/timeline-franchise/|future|Terminator timeline: the (extremely confusing) story so far|was Sarah always destined to give birth to a future resistance leader named John Connor, regardless of who his father might be?

http://www.telegraph.co.uk/film/terminator-genisys/timeline-franchise/|foster|Terminator timeline: the (extremely confusing) story so far|John Connor (Edward Furlong) is 10 years old and living with foster parents

https://en.wikipedia.org/wiki/Bulldog|patient|bulldog|Most have a friendly, patient nature. Bulldogs are recognized as excellent family pets because of their tendency to form strong bonds with children

https://en.wikipedia.org/wiki/Bulldog|short|bulldog|Bulldogs' lives are relatively short. At five to six years of age they start to show signs of aging.

https://en.wikipedia.org/wiki/Bulldog|peak|bulldog|Bull-baiting, along with bear-baiting, reached the peak of its popularity in England in the early 1800s until they were both made illegal by the Cruelty to Animals Act 1835

https://en.wikipedia.org/wiki/Bulldog|muzzle|bulldog|Though today's Bulldog looks tough, he cannot perform the job he was originally created for as he cannot withstand the rigors of running and being thrown by a bull, and also cannot grip with such a short muzzle

https://en.wikipedia.org/wiki/Bulldog|conditions|bulldog|Bulldog owners can keep these issues under control by staying aware and protecting their Bulldog(s) from these unsafe conditions.

http://www.factmonster.com/dk/science/encyclopedia/cameras.html|catches|cameras|A camera is a device that records pictures. It consists of a sealed box that catches the light rays given off by a source

http://www.factmonster.com/dk/science/encyclopedia/cameras.html|jelly|cameras|The emulsion consists of crystals of silver compounds in a jelly-like substance called gelatin.

http://www.factmonster.com/dk/science/encyclopedia/cameras.html|opposite|cameras|Colour films produce colour negatives, in which all the different colours in the image are replaced by their complementary, or opposite, colours. The dark colours appear as light areas and the light ones appear dark.

http://www.factmonster.com/dk/science/encyclopedia/cameras.html|reversed|cameras|It is a strange-looking version of the original scene in which dark and light areas are reversed.

http://www.factmonster.com/dk/science/encyclopedia/cameras.html|loaded|cameras|A digital image can be loaded into a computer, edited, printed out, sent by email, or stored on a website.

http://www.simplypsychology.org/memory.html|process|Stages of Memory Encoding Storage and Retrieval|Memory is the process of maintaining information over time

http://www.simplypsychology.org/memory.html|draw|Stages of Memory Encoding Storage and Retrieval|Memory is the means by which we draw on our past experiences in order to use this information in the present

http://www.simplypsychology.org/memory.html|aid|Stages of Memory Encoding Storage and Retrieval|Organizing information can help aid retrieval. You can organize information in sequences (such as alphabetically, by size or by time).

http://www.simplypsychology.org/memory.html|order|Stages of Memory Encoding Storage and Retrieval|If the doctor gives these instructions in the order which they must be carried out throughout the day (i.e. in sequence of time), this will help the patient remember them

http://www.simplypsychology.org/memory.html|recall|Stages of Memory Encoding Storage and Retrieval|Few, if any, people would attempt to memorize and recall a list of unconnected words in their daily lives.

Appendix B - Consent Form Pilot Test

Forespørsel om deltakelse i forskningsprosjektet

Ord betydning(Disambiguation)

Bakgrunn og formål

Dette studiet er en del av mastergradstudiet ved det Samfunnsvitenskapelige fakultet i Universitetet i Bergen. Formålet med denne studien er å samle informasjon til en sammenligning der man trenger menneskelig deltakelse. I masterstudiet mitt så blir det utviklet et program som skal velge et valgt ords betydning fra nettsider helst på samme nivå som et menneske. Resultatet fra studiet blir sammenlignet med resultatet fra det nåværende versjon av programmet.

Deltakelsen av personer i studiet er et tilfeldig utvalg av studenter som har hatt anledning til å delta.

Hva innebærer deltakelse i studien?

Dette studiet består av femten kryss av spørsmål på ca. 16 A4 ark, det vil omtrent ta 15 minutt av deltakers tid å fullføre. Spørsmålene vil omhandle generiske ord fra kontekster som deltaker skal krysse av en av de gitte betydningene som deltaker mener er riktig.

Hva skjer med informasjonen om deg?

Alle personopplysninger vil bli behandlet konfidensielt. Studiet vil ikke spørre om Personopplysninger siden personlig data ikke er relevant. All data vil bli kvantifisert og brukt til sammenligning.

Prosjektet skal etter planen avsluttes 01.12.2016

Data blir brukt i masteroppgaven og lagret deretter for eventuell bevis under muntlig eksaminering av oppgaven.

Frivillig deltakelse

Det er frivillig å delta i studien, og du kan når som helst trekke ditt samtykke uten å oppgi noen grunn. Dersom du trekker deg, vil alle opplysninger om deg bli anonymisert.

Samtykke til deltakelse i studien

Jeg har mottatt informasjon om studien, og er villig til å delta

---

(Signert av prosjektdeltaker, dato)

Hvis spørsmål angående studiet bruk kontakt informasjonen under.

Master Veileder:
Csaba Veres Mail: cve021@infomedia.uib.no

Master student:
Andreas Sekkingstad Mail: ase@student.uib.no Tlf: 47812617

Appendix C - Consent Form: The Experiment

Consent Form

Background and purpose:

This study is a part of the master's degree at the social Sciences faculty of the University of Bergen. The goal of this study is to gather information to a comparison where one requires human participation. In my master thesis I have built algorithms that decides a particular words sense based on a context. Where the context is website text context, the data collected here is to see whether the algorithm does as well as a human would.

What does it mean to participate in this survey?

This survey consists of 100 questions divided into 20 parts. The questions are multiple choice and the survey will take about 30 minutes. The questions are senses from a target word which the user will choose which is correct based on the context.

Personal information:

Any personal information will be treated confidentially. The study will not be asking for any personal information since personal information is not relevant. All data will be quantified and used for comparison.

The thesis is projected to end 01.12.2016.

Data will be used in the master thesis and saved for eventual evidence during thesis presentation.

Voluntary participation:

Any participation in this survey is strictly voluntary, any participant can withdraw his/hers consent at any time without reporting any reason. If a participant should withdraw, any personal information will be anonymised.

Contact Information

Master Veileder:
Csaba Veres Mail: cve021@infomedia.uib.no

Master student:
Andreas Sekkingstad Mail: ase@student.uib.no Tlf: 47812617

Appendix D

Future Work

This section presents some of the elements I either did not have the time to develop or functionality I would like to extend the built software into. Such as a server with a website, making the algorithms available for other to test as well. The reason most of these elements are non-existing is because of the simple reason that they do not answer the research question. Other than that, they would have been interesting to implement in the future.

New WordNet

It is clear that there is modest agreement in the second survey. The interesting fact beside the low agreement score is that 95 percent of the top two answers are fine grained, in a way that they are almost interchangeable, or at the least highly difficult to distinguish from one another. Which supports the hypothesis that a lot of the fine granularity of senses are so closely related that one could combine the two to one or remove one of them. Reviewing the top two survey answers using WordNet's definitions. Though this is a sample of just 100 different disambiguation answers from 100 different sentences, it is cause to investigate deeper into the phenomenon. Multiple times during this thesis are mentioned the difficulties when facing the fine grained definitions within the WordNet lexicology. Difficulties in distinguishing the senses from another when they both are closely related under the umbrella of a word. Proposed in this thesis are suggestions in how to avoid the cases where polysemy and homographs pose big problems for WSD and humans. When reviewing some of the cases where the users disagreed heavily, it still is difficult if not impossible to explain the difference between meanings. This problem exists whether there are a low or a high count of senses tied to a word. In conclusion, a revised or a new version of WordNet where closely related senses (polysemous instances) could be combined into one sense, would help WSD algorithms and users of WordNet in general.

Semantic Lifting

After running the program and retrieving the senses from different sources. A great opportunity arises to lift the data semantically into RDF triples, giving the opportunity to tag words with actual websites that is connected to the previously mentioned sources.

Web server

For the disambiguation software, it would be preferable to have a web server available for online use. Giving users the ability to copy the link of a website and retrieve the Websense of that URL. Or, if some user wants to try out the disambiguator on single sentences. Both these functionalities would help promote this thesis' results. There are most likely interested people who is interested in the domain of WSD, and if launching a website can help others research, it would maybe be worth the extra time. This means building a webserver and buying a domain for the website. I do not have the time available before the project is due., So in future work of the project it would be preferable to have the mentioned functionality.

Further Work on the Algorithms

As mentioned earlier, the algorithms built in the thesis are based on single words, and not combinations of more than one word. Unfortunately, this rules some of the concepts in WordNet out from the disambiguation if there is a combination within the context. Fixing this problem in the future would give the programs an edge when finding senses to target words, making them more complete, and fool-proof.

Increased testing

The lines acquired for the testing phase in this thesis are adequate, still, maybe if more testing lines were available it could give more weight to the results. On the other hand, the same lines have been given to humans. So, even though more thorough testing would be preferable, I need to give the same test to human subjects. And, having such a huge test can be bad for the human side of the testing. The subjects are more likely to get bored with the test and hurry through it rather than taking their time and being thorough. So keeping the number at one hundred lines should be enough.

Appendix E

Tools

This section presents the tools used in the development and writing the thesis. Since the funding has been limited, I have depended on using open source programs as much as possible. Below is a list and short explanation of the software, followed by the motivation for using said software.

Eclipse IDE

For the programming part of the project, I will most likely be using eclipse as the code editor. Eclipse is an Integrated Developing environment that will help speed up the coding process. I have experience with Eclipse, so learning new functions with this tool, will not take too much time.

Sublime Text 2

Simple but powerful code editor, millions of downloadable code snippets and plugins available for developing in any language possible. Convenient when developing HTML documents.

WordNet 3.1

A relational lexical database where words and senses are linked to eachother in some semantic way (hyponyms, hypernyms, co-occurrences, etc.). Popularly used in Natural Language Research, also a great tool for general lexical use.

Mendeley

Referencing tool, with functionality that enables users to add bibliography directly from internet sources directly to the online mendeley account. There is also software available for desktop use, complete with plugins for Open office and Word.

Word

High Level text editor for Windows machines. I have experience with this tool from previous classes, so using Word efficiently without too much practice should prove useful timewise.

NLTK

From their website *"NLTK is a leading platform for building Python programs to work with human language data".*(NLTK 2015). This explains the python package well, it has not only a module for disambiguation. But also for a lot of other functions, examples of such functions are:

-Part-Of-Speech Tagging
-Building sentence trees
-Classifiers & clusters
-Basic search methods within WordNet

WordNet::SenseRelate

A module built in the Perl programming language, has the ability to disambiguate senses from their respected contexts. The module has different distributions (see Development stage 1), able to either disambiguate entire sentences, word based from the context, or one target word in a context. The program is built on the WordNet Perl module as well, so coordination between synsets and Word part of speech elements is quick to grasp.

Python

Python is a high-level programming language, created in the 1980s. Python 3.0 is the version released in 2008, and was referred to as py3k during development. Python is a multi-paradigm programming language, it supports object-oriented programming and structured programming, and with extensions more paradigms can be used. It is highly readable with English key words. The use of the python is mainly in conjunction with NLTK.

Perl

Perl is a programming language designed and developed by Larry Wall, though not an official acronym, the best known backronym is “Practical Extraction and Reporting Language”. The programming language comes with different features like complex data structures and object-oriented etc. The Perl modules WordNet::SenseRelate and the WordNet::Similarity is what pointed me in the direction of the programming language.

R-Studio

R-Studio is an integrated development environment for the R statistics language, includes a code editor and includes R- packages from an easy install window and etc. The software is open source and commercial. Used specifically in this project for data investigation, and statistics.