2010-07-08

Utilizing Human-Computer Interactions to Improve Text Annotation

Marc A. Carmen
Brigham Young University - Provo

Utilizing Human-Computer Interactions to Improve Text Annotation

Marc Armstrong Carmen

A project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Arts

Alan K. Melby, Chair
Deryle W. Lonsdale
Eric K. Ringger

Department of Linguistics and English Language

Brigham Young University

August 2010

ABSTRACT

Utilizing Human-Computer Interactions to Improve Text Annotation

Marc Armstrong Carmen

Department of Linguistics and English Language

Master of Arts

The need for annotated corpora in a variety of different types of research grows constantly. Unfortunately creating annotated corpora is frequently cost-prohibitive due the number of person-hours required to create the corpus. This project investigates one solution that helps to reduce the cost of creating annotated corpora through the use of a new user interface which includes a specially built framework and component for annotating part-of-speech information and the implementation of a dictionary.

This project reports on a user study performed to determine the effect of dictionaries with different levels of coverage on a part-of-speech annotation task. Based on a pilot study with thirty-three participants the analysis shows that a part-of-speech tag dictionary with greater than or equal to 60% coverage helps to improve the time required to complete the part-of-speech annotation task while maintaining high levels of accuracy.

Keywords: part-of-speech annotation, user study, CCASH, active learning, cost-reduction

ACKNOWLEDGEMENTS

I would first like to thank my wife, Jennifer, and my beautiful children for being patient with me for all of these years. I am very grateful for the CCASH development team. I am especially grateful for Paul Felt and Owen Merkling for the time and effort they put in to the development project and to Robbie Haertel for his guidance and direction on this project.

I am also very grateful for my committee Eric Ringger, Deryle Lonsdale, and Alan Melby. Their guidance and patience over the past few years has been especially helpful and appreciated.

Table of Contents

List of Tables

List of Figures

1. Introduction

There has been significant discussion within the linguistics community for many years regarding the best approach to analyzing linguistic phenomena. One school of thought is that it is best to use the intuition of a native speaker to find and analyze phenomena. The second school of thought is that it is best to use observed language—written or spoken—to analyze phenomena. One of the obstacles preventing the implementation of the second strategy is the requirement for a significant amount of observed language data to analyze. This may not sound like a particularly complicated task; however, compiling and creating useful data is more complicated than it seems. In addition, multi-faceted corpora—those that can be used across many different areas of research—are even more difficult to compile. Recent advances in computer technology (i.e., faster processors, more storage space, more memory, etc.) have helped to reduce the complexity of corpus compilation and analysis. However, many problems still remain in the process of creating a corpus.

The most basic form of a corpus is a collection of words, phrases, sentences or documents that are stored in a format that enables searching and analysis. However, corpora are only as useful as the information they contain; any additional information that can be added to a corpus enhances the ability of researchers to perform analysis on a corpus. Most corpora that are created today at least include or are divided into domains (i.e., newspaper, academic, technical, literary, etc.). Depending on the goals of the research, a selection of domains may all be compiled into one corpus but typically the domain will be specified as extra information known as metadata. Knowing the domain of a set of documents is important and allows for a detailed analysis of that domain or potentially even a cross-comparison of different domains. Even though the domain is an important piece of information for a corpus there is more information that can be gathered about a corpus. For example, a corpus can contain phonological, morphological, syntactic, and semantic information which allows for more in-depth analysis. A simple example is to compare the adjectives that occur before the words “man” and “woman” in an English corpus. This can be done with a corpus, such as the Corpus of Contemporary American English (COCA)[1] which has been compiled by Dr. Mark Davies at Brigham Young University. COCA, which was released in 2008, contains 385 million words which have all been annotated for part-of-speech using the CLAWS-7 tagger (Davies 2009).

The Penn Treebank was one of the earliest examples of large-scale annotated corpora. The Penn Treebank began as a project in 1989 and, after three years of work, the team had annotated 4.5 million words of American English (Marcus, Santorini, and Marcinkiewicz 1994). The Penn Treebank includes both part-of-speech (POS) and syntactic information. The annotation process was done by first automatically annotating the data using a variety of computer algorithms created for POS annotation, which achieved an error rate of 2-6%, and then human participants corrected or confirmed the automatic annotations. The human annotators used a program embedded into the GNU Emacs editor. After a month of training, the annotators were correcting the annotations at speeds faster than 3,000 words per hour. Since its inception, the Penn Treebank has been utilized in a variety of studies and projects including annotation of morphology, syntax, and semantics and in numerous projects in the fields of computer science and linguistics. The utility of the Penn Treebank is evident from the number of references

[1] http://www.americancorpus.org/

(1,779) those keywords generate in a search of CiteSeerX[2], a commonly used search engine for scientific information.

Unfortunately manually annotating or even correcting 4.5 million words can be cost-prohibitive. Without sufficient funds, most research projects will only be able to annotate a small subset of this data. On top of that, some languages are known by so few individuals that the number of participants in the annotation process is limited. Due to the potentially cost-prohibitive nature of manual corpus annotation there is substantial research regarding methods to reduce the cost of annotation. This project focuses on English part-of-speech (POS) annotation and proposes using a POS tag dictionary to reduce the cost of corpus annotation while maintaining high levels of accuracy. Utilizing a dictionary with significant coverage in the annotation process should reduce the cost of corpus annotation while maintaining high levels of accuracy.

Chapter 2 in this report will discuss some of the previous work related to cost-reduction of corpus annotation including software tools that have been created as well as different computational algorithms. Chapter 3 discusses the design of CCASH (Cost-Conscious Annotation Supervised by Humans) and the requirements that were considered based on existing tools. An overview of the user study performed for this project is discussed in detail in chapter 4 followed by the results of the study in chapter 5. Chapters 6 and 7 will provide a brief conclusion and the possible future work that stems from this project.

[2] http://citeseerx.ist.psu.edu/

2. Previous Work

There are different ways of approaching text annotation. The first is to hire a group of researchers to go through and manually annotate the data. However, the resulting annotation depends on the researchers' proficiency in the language and their linguistic analytical skills. In addition, the cost incurred by hiring these individuals would be immense if the goal were to annotate the mega-corpora being produced today. A second approach to annotation is to create a computer algorithm that performs the annotation task. Computer algorithms can often perform the annotation task at or near the same level of accuracy as human annotators. These types of algorithms include supervised and semi-supervised algorithms. A supervised algorithm is an algorithm where a model is created and fit to a set of training data. A semi-supervised algorithm is similar to a supervised algorithm but it uses un-annotated data as well as annotated data for training. For example, a modern statistical tagging algorithm can usually achieve around 97% accuracy on an English part-of-speech task which is just below the 98% tag accuracy in the Wall Street Journal portion of the Penn Treebank. Although utilizing a computer algorithm for the annotation process does require a developer or development team, the cost of time and money is minimized because they are usually implementing an existing algorithm. This process also benefits from an increasing number of software libraries. Due to the fact that statistical approaches still require significant amounts of annotated data, they truly only reduce the cost of corpora that are annotated using a statistical algorithm that is trained on an existing corpus that has been annotated. This means that a purely statistical approach only reduces the cost of corpus annotation after a significant amount of data has been annotated. A third, hybrid form for the annotation process involves human-computer interaction. One form of human-computer interaction involves a statistical algorithm known as active learning. Active learning is similar to a supervised algorithm because it uses previously annotated data to train the algorithm. However, it is also similar to a semi-supervised algorithm because it makes use of un-annotated data. Unlike a semi-supervised algorithm active learning uses an oracle, which can be a human, another program, or something else that has knowledge regarding the task, that provides feedback to the active learning algorithm. Once the active learning algorithm has received feedback from the oracle it adjusts the statistical model accordingly.

2.1. Human-Computer Interaction

There are different ways of utilizing humans and computers together for text annotation. The most basic is to utilize a software tool that allows humans to annotate a corpus. Another example is using a software tool to perform the initial annotation, which is then corrected by humans. Some of these methods overlap with each other but it is important to make note of these different approaches. For example, the Penn Treebank, according to Marcus, Santorini, and Marcinkiewicz (1994), was annotated first using a variety of computer annotators. Then during the second stage of the process human annotators used “a mouse-based package written in GNU Emacs Lisp” which allowed annotators to select a tag and change it if necessary. The software would then check the entry against the list of legal tags. A more recent experiment by Fort and Sagot (2010) used software to pre-annotate the corpus and found that pre-annotating the data can increase the quality of annotation for both accuracy and inter-annotator agreement. However, Fort and Sagot also made an important observation that this method of corpus development can lead to biases that must be identified so that the annotators can correct the pre-annotation accurately. The final piece of information that Fort and Sagot found was that even using a small corpus for training and pre-annotating can improve the speed of annotation. For example, they found that only training the tagger on 50 sentences does not yield a highly accurate tagger but does speed up the annotation process. Although using computer algorithms for pre-annotation is an important facet of performing text annotation, the remainder of this chapter will focus on currently available software tools used for text annotation and active learning.

2.2. Software Tools

The Emacs tool created for the annotation of the Penn Treebank is one example of available text annotation software. Knowtator (a plug-in to Protégé, an ontology editor), which is written in Java, allows the easy creation of complex annotation schemas (Ogren 2006). Ogren points out that although other existing tools come with a variety of tasks available out-of-the-box it can be difficult to extend the functionality of existing packages to a customized annotation task. However, because Knowtator extends the functionality of Protégé it has access to the existing user interfaces to help with creating the annotation schema. This means that Knowtator's annotation schema can be applied to an annotation task without having to write any additional software but rather only creating an annotation schema and then applying it to the task. WordFreak, another Java application, is an extensible annotation system that allows for easy integration of additional components and new annotation tasks (Morton and LaCivita 2003). In addition, it provides access to several automatic tools including sentence detectors, part-of-speech taggers, and parsers. Moreover, the development team is actively working on including other open source annotators as plug-ins to WordFreak. GATE is another Java based tool that was begun in January of 1995. Over the years the GATE team has put together a variety of components that focus on language engineering and can be used, extended, and customized to fit the needs of a particular task (Cunningham et al. 2002). GATE divides the annotation process into three components: language resources, processing resources, and visual resources. This allows a project to use language resources like lexicons and corpora along with existing algorithms, which are processing resources, to help in the annotation process. Finally, it includes prebuilt and extensible visual resources, or graphical user interfaces for working with the annotation process. In addition, other components such as GATE TeamWare and GATE Cloud allow for the annotation process to be distributed to multiple annotators in different locations. Finally, the Jena Annotation Environment (JANE) is a tool that specifies a project as a set of documents to be annotated and an annotation scheme (Tomanek, Wermter, and Hahn 2007). Unlike the other tools that have been discussed, JANE includes a semi-supervised component called active learning that allows for the machine algorithms to be improved based on human input.

2.3. Machine Learning

As has been mentioned previously, a common way to reduce the cost of text annotation is to use computer algorithms to perform the task. Brill (1992) reported on a rule-based part-of-speech (POS) tagger. Initially this tagger assigns the most likely tag for a word based on the training data. If a word was not seen in the training data, then it uses a set of rules to annotate the word; finally, if the rules don’t match, then the word is simply assigned the most common tag in the corpus. The output of the initial tagger is compared against another part of the corpus to find common errors. Utilizing the error information and the contextual information the tagger will correct erroneous annotations. Brants (2000) reported on a statistical part-of-speech tagger that used information regarding a word and the two words before it (known as a trigram) as well as statistical information gained from the training corpus to determine the tag of a word. This algorithm achieved around 97% accuracy which is comparable to manually annotated corpora. Most POS taggers that are used today for English achieve approximately 97-98% accuracy. However, the majority of these algorithms require already existing annotated data and their high accuracy is directly correlated with the amount and quality of training data that is provided to the computer algorithm. This is not a problem with a task like English part-of-speech tagging because there are sufficient amounts of data to utilize for the training process. Unfortunately there are significantly fewer sets of training data for projects dealing with other tasks like named entity recognition or sentence parsing. In addition, lesser-resourced languages—those languages that have few existing resources available—are even more problematic because there is little or no training data available. Of the many corpora available today, few of them have been annotated and even fewer are available for languages that are less common. One type of algorithm that can be used to reduce the amount of training data that is required is active learning.

Active learning is a machine learning approach that can reduce the amount of required training data and therefore reduce the cost of the annotation project. Active learning is similar to other machine learning algorithms because the amount and quality of the training data affects the accuracy of the algorithm. However, active learning uses a method to determine which pieces of the corpus will contain the most information. Once a chunk of text has been selected as the most valuable a human annotator, or oracle, annotates the text providing the algorithm with newly learned information. Using the newly learned information, the statistical algorithm is adjusted and reapplied to the annotation task. As more data is provided to the algorithm, it becomes more selective about which sentences contain the most useful information and the more improved the algorithm becomes. Over time the algorithm should be able to achieve the same levels of accuracy as other machine learning algorithms that have trained on the entire set of data but it should occur more quickly and with much less work than is required to create a hand-annotated corpus.

Although there is no guarantee that active learning will reduce the total cost of producing an annotated corpus, some user studies have reported improvements in either time or accuracy during the annotation process. Ringger et al. (2008) conducted a user study which allowed the authors to define a cost model for time required for English POS annotation with the aid of active learning. The authors presented predictive linear cost models for both word-at-a-time and sentence-at-a-time active learning-based annotation. Palmer, Moon, and Baldridge (2009) conducted a user study involving automatic pre-annotation and active learning with both an expert and non-expert annotating the Uspanteko language. They found that machine labeling and active learning can increase the accuracy of human annotators but the degree to which they increase the accuracy is related to the experience and knowledge of the annotators. Although there is research that shows that active learning reduces the total amount of time and money required to create a corpus, my project did not make use of active learning. Instead my project concentrated on the effects of utilizing dictionaries to improve annotations by humans and the effects and implementation of an active learning algorithm were out of scope. However, utilizing active learning in tandem with a dictionary is a novel idea and the potential future work will be addressed in the final chapter of this report.

3. Project Design

Despite the availability of multiple annotation software tools, there is a growing need for annotated corpora. Unfortunately the creation of annotated corpora is often cost-prohibitive. As a result, a considerable amount of current research works towards lowering the costs of creating annotated corpora. The goal of this project is to determine the effect of a part-of-speech (POS) dictionary on the POS annotation process. It was determined that the best way to analyze the effects of a dictionary on the annotation process was to create a software tool and working with the Natural Language Processing (NLP) group at Brigham Young University (BYU) a set of requirements was developed for the tool that would be utilized in these experiments. These requirements stemmed from research and experience with some of the tools and methods previously discussed. The requirements are that the system must:

- allow developers to implement proven cost-efficient annotation methods
- allow developers to implement new cost-efficient annotation methods
- facilitate exploratory studies and the comparisons of annotation methods
- allow for custom annotation tasks including, but not limited to, part-of-speech tagging
- coordinate the efforts of multiple annotators

The remainder of this section will discuss CCASH (Cost-Conscious Annotation Supervised by Humans) and how it achieves the five requirements mentioned above.

3.1. CCASH Framework

One of the most basic questions facing a software developer is where to store the data and how to allow end-users to access the data. Most Internet users have become accustomed to web-based applications like webmail (Gmail, Microsoft Live, Yahoo!, etc.), online productivity packages (Google Docs, Zoho, Adobe Acrobat.com, etc.), and web searching (Google, Yahoo!, Microsoft Bing, etc.); as a result, a web-based interface would be ideal for most annotators to work with. A variety of technologies can be used to build Internet applications. The CCASH development team decided to use the Google Web Toolkit (GWT) for several reasons. First, GWT comes with many extensible components that can be implemented right out of the box and allow developers to create novel components with less work. For example, for the part-of-speech (POS) annotation task the development team created a component that filtered a list of possible POS tags based on what was typed into a text box. In addition, GWT allows developers to write the software in Java, which many software developers are already familiar with, and then compiles into JavaScript which is cross-compatible with many different browsers. While JavaScript is not rendered the same in every browser, GWT's compiler provides one of the best cross-browser experiences. GWT allowed the development team to speedily complete a web-based interface similar to interfaces that most Internet users are already familiar with. Using a web-based interface reduces the learning curve due to user interface design but still allows the software to distribute and coordinate the work between multiple annotators.

Many research tools and algorithms today are implemented in Java. In addition, the work previously done by the NLP research group was almost completely Java-based. Utilizing GWT allows for the development team to make use of existing methods and algorithms for reducing the cost of annotation in the project. Moreover, as students and researchers implement new components, whether they are a backend component for processing the data or a user interface component, it is relatively simple to integrate with CCASH. This extensibility allows CCASH to be more like a framework which allows for more future research to be completed along the same lines.

A MySQL database was used on the backend for storing any information for the CCASH application. This includes any data that needs to be annotated or has been annotated as well as dictionaries and other lexical resources. In addition, using a relational database as the storage component of the application allows easy storage of metrics including time and accuracy. Although these metrics could be stored in other formats such as XML, a relational database allows the software to store and retrieve data in an organized fashion. Using a database in a situation like this improves the speed of the software as the libraries that are used to connect to the database have been honed and improved strictly for that purpose. In addition, using the Java and MySQL technologies together allowed the development team to implement a database persistence component using Hibernate. Database persistence allows the development team to spend more time on software development and less time on database architecture and development. In a traditional application, a developer would need to handcraft any queries used to communicate with the database. Database persistence allows the development team to hand those complicated queries over to the persistence engine thus reducing the overall development time.

The CCASH framework created by the development team allows for the rapid and easy creation of new interfaces and components to expand the scope and ability of the system. For example, while the POS annotation interface was being developed, another developer simultaneously developed an interface used for named entity annotation. This interface was implemented utilizing the same framework and could easily be put in place of the POS annotation interface. An ideal system would allow different annotation projects to utilize different components and user interfaces without having to make code changes or restart the application server.

Although there were many advantages to using GWT for CCASH development it was not without its difficulties. While working on the POS annotation component we initially had problems because the sizing wouldn't work as expected. In addition, because the system was compiled from Java into JavaScript there were some instances in which a component would or would not work correctly using the GWT test environment and the opposite would be true in a live browser situation. The development team quickly learned that the GWT test environment was not to be trusted completely. Overall though, the use of GWT was advantageous to the CCASH project and this user study.

3.2. Contributions

The ultimate implementation of an entire annotation system was the work of many members of NLP research group. I designed the original database for the dictionary system and the dictionary data objects. I also helped create the POS annotation interface used for this project. There were two team members that were primarily responsible for putting together the CCASH framework and a third team member helped to design and implement the rest of the database and data objects. Other members of the NLP research group consulted on the design and helped with the testing of the system. I was directly responsible for designing the user study, preparing the data, and analyzing the results of the user study. This study and its results are discussed further in the following chapters. The CCASH software has been licensed as an open source project and is available on SourceForge at http://sourceforge.net/projects/ccash/. Although the first phase of software development—which includes the framework and database—is complete there is still active development on the project including a user interface for Syriac—an ancient Semitic language of Syria which is today used primarily for liturgical purpose for Syrian Christians.—part-of-speech tagging and an administration interface to actually manage the annotation projects.

4. User study

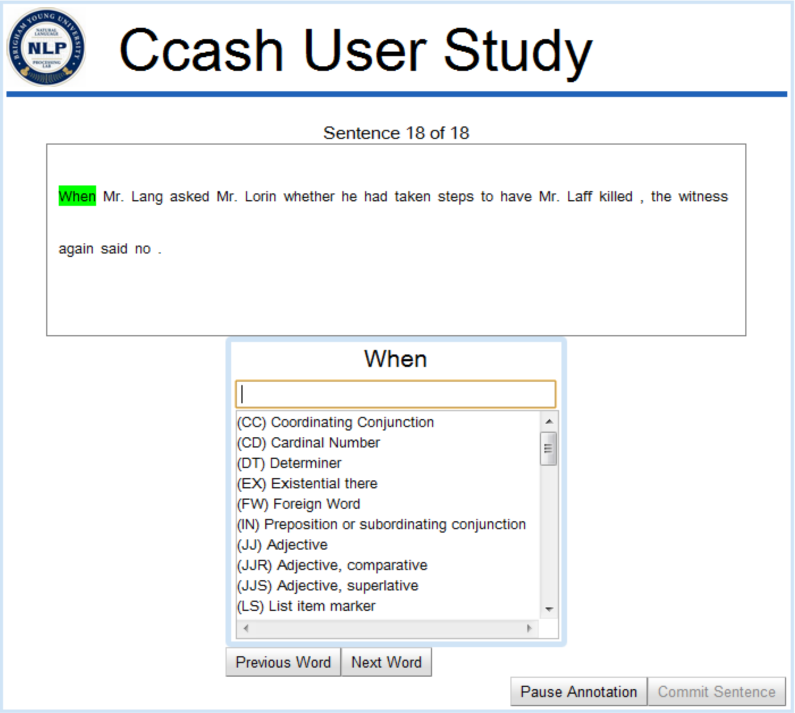
4.1. Design

This study consisted of thirty-three participants annotating eighteen English sentences randomly selected from the Penn Treebank. Even though there is sufficient existing data for a statistical approach to English part-of-speech (POS) annotation, English was selected as the language for this study for several reasons. First, the results of this study will be examined using a statistical analysis which requires more data than could easily be found for a lesser-resourced language. Although it is possible to find thirty-three participants to help with a user study for other languages it was quickest and easiest to find participants for an English user study. Second, the Penn Treebank provides an existing gold standard set of data that can be used for preparing the user study and analyzing the results. Finally, as mentioned in chapter 1, it is common in natural language processing and computational linguistics to use the Penn Treebank for POS tagging, which means that the results of this user study are comparable to prior studies. During the user study every user was presented with the same list of sentences in the same order but with varying levels of dictionary coverage. Dictionary coverage is defined by the following formula:

$$coverage = \frac{\#\ of\ tokens\ with\ at\ least\ one\ dictionary\ entry}{\#\ of\ tokens\ in\ the\ sentence}$$

In POS annotation, a token is an individual item that will be annotated. The size of the token varies depending on the granularity of the project and the language. For an English annotation project, the token will nearly always be an individual word. The manner of data selection and dictionary creation will be discussed in more detail in this chapter.

For this user study a POS annotation widget was created for the CCASH framework. Figure 1 shows the English POS annotation widget that was used for this user study. This widget presents the user with the sentence to be annotated and automatically selects the first token in the sentence which can be seen in Figure 1 directly under the title of the page. The token that is currently being annotated is highlighted in the sentence in green and presented in the annotation portion of the user interface. The annotation part of the user interface also contains an auto-complete box and a list of possible tags. As the user enters text into the auto-complete box, the list of possible tags is filtered accordingly. If a particular sentence for a participant is assigned a dictionary with coverage greater than 0% then the initial list of possible tags is filtered according to the contents of the dictionary. When a dictionary with coverage less than 100% is in use for a sentence some tokens will initially be presented with the filtered dictionary list of tags while other tokens will still be presented with the complete list of possible tags.



Sentence 18 of 18

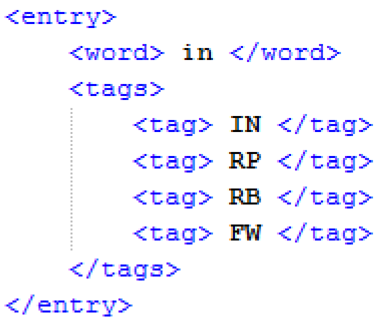
When Mr. Lang asked Mr. Lorin whether he had taken steps to have Mr. Laff killed , the witness again said no .

When

Figure 1 CCASH POS Annotation Interface

To avoid human error or bias in the dictionary creation, an automatic method was employed to create the dictionaries offline with 20%, 40%, 60%, and 80% coverage. To create the dictionaries, eighteen sentences were randomly selected for annotation by each participant as well as four sentences for the training portion of the user study. The sentences were either short (12 tokens), medium (23 tokens), or long (36 tokens). The exact sentence lengths were determined by first creating a list of each of the sentence lengths and then sorting them in ascending order. Finally, the list of lengths was split into thirds which provided the maximum length for both the short and medium length sentences. Each sentence was then assigned to a category according to its length; finally, the mean of each set of sentences was calculated giving the specific lengths used in this study. Those twenty-two sentences—eighteen training and four tutorial—were removed from the corpus and the remaining sentences were randomly shuffled and split in half into a set of training data and held-out data.

Next, a base dictionary for each dictionary coverage level (20%, 40%, 60%, and 80%) was created by iterating over each sentence in the training data and adding each token and its POS tag in the sentence to the dictionary. The tokens for words from sentences in the corpus were added to each dictionary until the desired coverage level, which was calculated using the held-out data set, was reached. The two exceptions to this process were the 0% coverage dictionary, which contained no entries in the dictionary, and the 100% coverage dictionary, which was built using the entire set of training data. With a base dictionary for each coverage level complete, a new dictionary for each sentence and coverage level was created by either adding or removing sentences until the approximate desired coverage level was reached. Using this process to create the dictionaries helped to ensure that each dictionary was as close as possible to its desired coverage level. On average the dictionaries were within 2.12% of the desired coverage level. Once the dictionaries were complete, the data was written out to XML files for offline usage.   Storing the data in the intermediate XML format allowed the database to be reset and modified and the application tested without having to regenerate any information. This meant that each time the application was restarted and restored the XML data could easily be parsed and stored in the database for use in the application. Figure 2 below shows an example of an entry for the word “in” in a tag dictionary with 80% coverage. A more detailed example of the dictionary XML files can be found in Appendix B.



```
<entry>
    <word> in </word>
    <tags>
        <tag> IN </tag>
        <tag> RP </tag>
        <tag> RB </tag>
        <tag> FW </tag>
    </tags>
</entry>
```

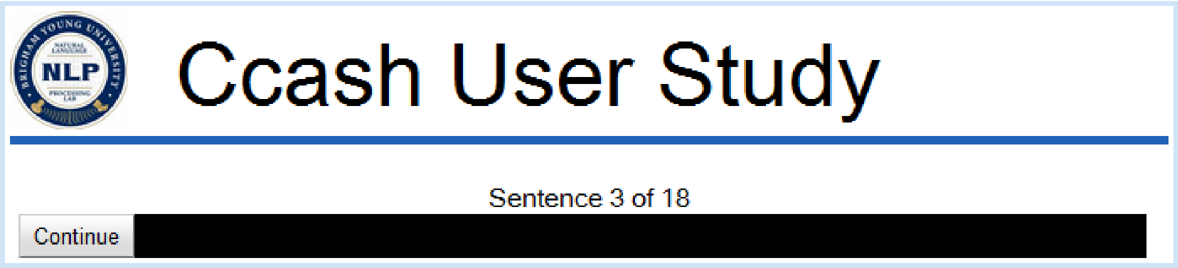
Figure 2 Sample of the dictionary in XML format for the word “in”

4.2. Participants

The thirty-three participants were first-year linguistics graduate students in a required syntax and morphology course. Questionnaires were given to the participants before and after the study which included questions regarding previous coursework that included part-of-speech (POS) tagging, the participant’s native language, and estimations of how well they performed the annotation task. Twenty-three of the participants are native English speakers, and over 50% of the students had taken one or fewer previous courses that included POS tagging. In addition, when asked about their tagging proficiency, over 50% of the participants rated themselves with a 1 (lowest proficiency) or 2 out of 5 (highest). The students were given an assignment by their instructor and were told that credit would be given based only on completion of the study and whether or not the results indicated that the participant had taken the study seriously: participants were informed that both accuracy and time were important for the study. Subjects were allowed about two weeks to complete the study on their own time. On the day of the assignment they were provided a sheet of paper containing the instructions on how to find the study, a list of possible tags, and examples for each tag. A copy of the sheet provided to each student is found in Appendix A.

4.3. Implementation

When a participant went to the website for the user study they were welcomed with a set of instructions. Those instructions informed them that the purpose of the user study was to measure both the time required to complete the task and the participant's accuracy on the task. The participants were then asked to remove any distractions so that they could complete the user study to the best of their ability. Due to the fact that the timing information was so important for the study the participants were given instructions regarding the "Pause Annotation" button which was implemented in the POS annotation widget and allows the user to click on a button to stop the time if necessary while performing the task. When the "Pause Annotation button is pushed the sentence and any data on the screen are removed to prevent potential cheating and maintain the accuracy of the study as is seen in Figure 3. When the user clicked the "Continue" button



Ccash User Study

Sentence 3 of 18

Figure 3 CCASH "Pause" screen to allow for accurate timing and prevent cheating

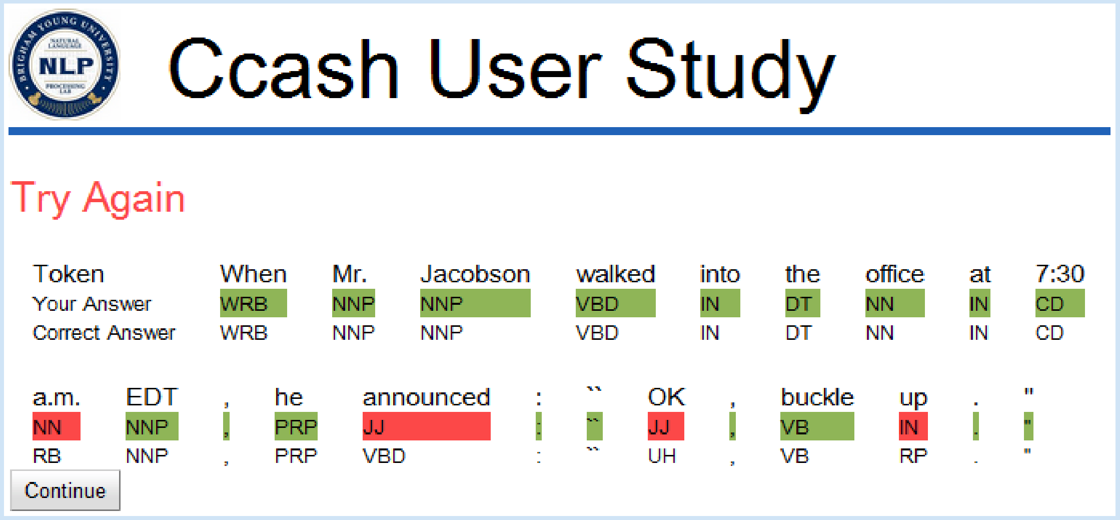
from the paused screen, they were allowed to continue from where they left off. Introducing the "Pause" button allowed us to track how much time the participant was actually spending on each sentence assuming they used the pause button when appropriate.

The remainder of the instructions provided an overview to the user interface and how to work with CCASH. After the participant clicked the "Continue" button on the main instructions page they were presented with the first questionnaire that asked:

1. Are you a native English speaker?
2. Have you participated in this study before?
3. How proficient are you at tagging?
4. How many previous classes have you taken that have discussed part-of-speech annotation?

Additional information gathered from this questionnaire will be presented with the final results.

After the questionnaire the participants began the tutorial. Every participant was shown the same four tutorial sentences in the same exact order with the same level of dictionary coverage for each sentence. The purpose of the tutorial sentences was two-fold. First, it allowed the user to learn and become accustomed to the user interface, reducing the learning curve of the system and



Ccash User Study

Try Again

Token	When	Mr.	Jacobson	walked	into	the	office	at	7:30
Your Answer	WRB	NNP	NNP	VBD	IN	DT	NN	IN	CD
Correct Answer	WRB	NNP	NNP	VBD	IN	DT	NN	IN	CD

a.m.	EDT	,	he	announced	:	``	OK	,	buckle	up	.	''
NN	NNP	,	PRP	JJ	:	``	JJ	,	VB	IN	.	''
RB	NNP	,	PRP	VBD	:	``	UH	,	VB	RP	.	''

Figure 4 A sample tutorial sentence with corrections

therefore the variance of the results due to the user interface. Second, the tutorial sentences helped to familiarize the participants with the Penn Treebank list of tags. After the participants annotated a tutorial sentence they were shown their results as is seen in Figure 4.

Subjects were expected to repeat each tutorial sentence until they were able to annotate all but one token successfully. The one token leniency was simply to allow for human error in the tutorial. Recording of time and accuracy did not begin until after the tutorial sentences were completed, so the final results were not affected directly by time and answers on the tutorial sentences.   However, this prevented us from determining exactly how long the entire user study lasted which could have been beneficial in the final analysis. If the user study is ever repeated, tutorial time and accuracy should be recorded as an independent data set. This would allow for further analysis regarding accuracy and time with the fatigue of the participant.

Once the participants finished the tutorial they were reminded to remove all distractions before they began the main part of the user study. The user interface of the main study was exactly the same as it was for the tutorial sentences except that participants were not shown their results following each sentence. The participant was presented with a sentence to annotate and either the entire list of tags or a partial, filtered list based on the level of dictionary coverage they were assigned for that sentence. If a dictionary contained a sufficiently complete tag inventory for a given token, the limited options for that token made the choice potentially easier for the annotator. If a dictionary entry was missing from the list, the annotator could add that option to the dictionary. The trade-off is that although a dictionary has the potential to accelerate annotation, an incomplete dictionary may require additional effort to augment. This is particularly the case in this user interface, since the user must click the “Select Different Tag” button, as seen in Figure 1, and choose from the complete list of tags when the desired option was not in the initial filtered list. It is possible that the tag dictionaries could affect the participant’s decision regarding a tag in a negative way: a complacent annotator may choose an option simply because it is the best in the list rather than considering the full range of options. To discourage that level of over-reliance on the dictionaries, none of the instructions provided to the participants described the list of tags as a dictionary but rather a list of suggestions. It was believed that this semantic difference would prevent some of the participants from putting complete trust in the dictionaries.

After completing all eighteen sentences the participants were presented with another short questionnaire that asked:

1. How accurately do you think you performed on this experiment?
2. Did you have the tag reference sheet by your computer while you did this study?
3. Did you pause as necessary during the annotation process to ensure accurate timing?

Participants were also given the opportunity to provide free-text feedback regarding the user study and its interface. Finally the participant was presented with a congratulatory message providing them with their serial number which was used to prove that the student finished the assignment and to make sure that only results from those students were included in the results used for this analysis.

5. Results

The results will be presented in three different sections. The first section reports on a statistical analysis concerning the role of dictionary coverage in annotation accuracy and speed. The second discusses the feedback received from end users. The final section of this chapter discusses a post-hoc analysis on the affect of the number of dictionary entries on the accuracy and time of annotation.

5.1. Statistical Analysis

As mentioned previously, thirty-three total students participated in the user study. Twenty-three of those students were native English speakers. Time and accuracy, both overall and sentence specific, were tracked for each participant. Time was measured in milliseconds according to events that occurred in the user interface. Measurements were taken when a new sentence was requested by the user interface, when the new sentence was presented to the user, when the annotation was completed, and any time the user paused or resumed the task. This allows us to construct a timeline and determine the total time spent annotating a sentence. Accuracy was determined by dividing the number of correctly annotated tokens by the total number of tokens in a sentence.

On average the participants performed the annotation task with 88.73% accuracy. The lowest accuracy was 80.52%, and the highest accuracy was 93.90% for the study. The non-native English speakers scored an average of 88.02% compared to the native speakers' 88.96%. The participants required from 22.76 minutes to 118.43 minutes to complete the study, with an average of 42.63 minutes. We do know from the participant feedback that some subjects did not always use the pause functionality of the user interface and, as a result, the higher times may not be completely accurate measures for the study. The non-native speakers took approximately 20 minutes longer than the native speakers to complete the study. Table 1 provides details on the speed and accuracy of the user study participants based on the answers they provided in the survey. In addition, there are some other important details revealed with these statistics. The tag reference sheet had little effect on the accuracy of the participants but did affect the times of the participants. Those with the tag reference spent more time on the annotation task, likely looking up information, but achieved similar accuracy scores. In addition, the individual that spent the most time on the study also reported that they appropriately used the pause button which means that the actual length of the study was even longer for that individual.

Table 1 Metrics for speed and accuracy of participants

		Min	**Median**	**Mean**	**Max**	**St Dev**
Accuracy	Native Speaker	84.04	88.97	88.89	93.90	2.87
	Non-Native Speaker	80.52	88.03	87.18	91.55	3.81
	Tag Reference	80.52	88.50	88.29	93.43	3.01
	No Tag Reference	81.46	89.44	88.67	93.90	4.20
	Appropriately Paused	80.52	89.44	88.56	93.90	3.31
	Not Appropriately Paused	85.45	85.92	86.46	88.03	1.38
Time (min)	Native Speaker	22.76	41.67	41.50	76.68	13.98
	Non-Native Speaker	31.37	63.26	65.27	118.43	30.61
	Tag Reference	22.76	43.41	52.30	118.43	24.39
	No Tag Reference	28.35	33.53	35.33	42.95	5.91
	Appropriately Paused	22.76	42.77	48.58	118.43	23.15
	Not Appropriately Paused	31.37	41.79	49.95	76.68	23.73

The most important results from this study concern the sentence-level statistics for each length/coverage level bucket. The baseline was the performance of the participants on each sentence-length bucket given a dictionary with 0% coverage (meaning all tag options were presented for each token). Consequently, the null hypothesis is that having no dictionary has the same effect on time and accuracy as having a dictionary. The time and accuracy for each sentence is analyzed using a standard t-test as well as a permutation test (Menke and Martinez 2004). The results were analyzed using both of these approaches because the t-test is the commonly used analysis for this type of comparison. However, the advantage of the permutation test is that it does not require the data to have a normal distribution. In the end, both analyses yielded similar results. Table 2 demonstrates the results of both time and accuracy given the length/coverage level buckets. The results show that as the level of dictionary coverage increased there was a significant improvement in both time and accuracy. For each sentence length, statistically significant improvement occurred when dictionary coverage was at or above 60% with a confidence level of 80% or higher; however, most of the results were achieved with a confidence level of 95% or higher. A dictionary with 100% coverage was nearly always optimal showing improvement with a confidence level of 99% for most sentence lengths. Although Table 2 is useful for seeing the exact values and highlighting those values that have high levels of confidence it is also useful important to visualize the overall trend of the result. This is easily seen in Figure 5 and Figure 6. Figure 5 shows that as the level of coverage increases the mean time of the participants decreases. This holds true for each length bucket that was examined. Figure 6 shows that as the level of coverage increases the accuracy of the participants increases which holds true for each length bucket as well.

Table 2 Sentence level results for each sentence-coverage level

The “Num” column indicates the number of data points available for the condition. “Perm” is analogous to p-val, but for the permutation test. Significant (at confidence level 90% or higher) results are highlighted

			Time					**Accuracy**				
Length	**Coverage**	**Num**	**Min**	**Mean**	**Max**	**p-val**	**Perm**	**Min**	**Mean**	**Max**	**p-val**	**Perm**
12	Full Dict	31	54	106	174	0.50	1.00	0.50	0.80	1.00	0.50	1.00
	20	31	48	136	238	0.79	0.41	0.58	0.81	1.00	0.43	0.87
	40	33	39	94	204	0.35	0.71	0.50	0.83	1.00	0.21	0.40
	60	29	40	100	139	**0.01**	**0.02**	0.67	0.83	1.00	0.18	0.37
	80	32	30	94	204	0.24	0.49	0.75	0.86	1.00	**0.00**	**0.01**
	100	31	26	85	133	**0.00**	**0.01**	0.75	0.86	1.00	**0.01**	**0.01**
23	Full Dict	27	64	258	264	0.50	1.00	0.70	0.87	1.00	0.50	1.00
	20	31	88	191	309	0.86	0.31	0.70	0.86	0.96	0.76	0.50
	40	29	88	191	253	0.22	0.44	0.74	0.88	1.00	0.30	0.62
	60	30	66	160	257	**0.07**	0.17	0.83	0.87	0.96	**0.08**	0.18
	80	30	54	130	225	**0.00**	**0.00**	0.83	0.89	0.96	**0.01**	**0.03**
	100	31	52	121	202	**0.00**	**0.00**	0.83	0.90	1.00	**0.06**	0.13
36	Full Dict	33	121	265	533	0.50	1.00	0.75	0.88	1.00	0.50	1.00
	20	32	113	248	465	0.15	0.32	0.72	0.87	0.97	0.71	0.57
	40	32	93	282	577	0.32	0.65	0.75	0.90	1.00	0.16	0.33
	60	30	82	219	353	**0.00**	**0.00**	0.81	0.92	0.97	**0.00**	**0.00**
	80	28	85	204	310	**0.00**	**0.00**	0.81	0.93	1.00	**0.00**	**0.00**
	100	31	90	191	318	**0.00**	**0.00**	0.78	0.93	1.00	**0.00**	**0.00**

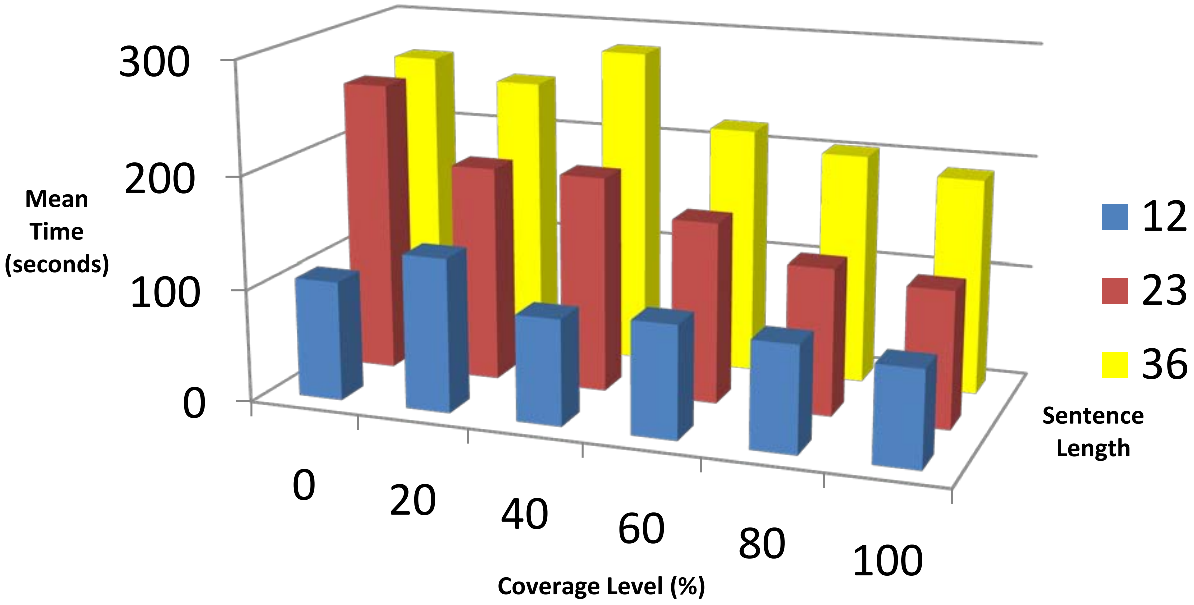


Figure 5 Impact of tag dictionaries on Time

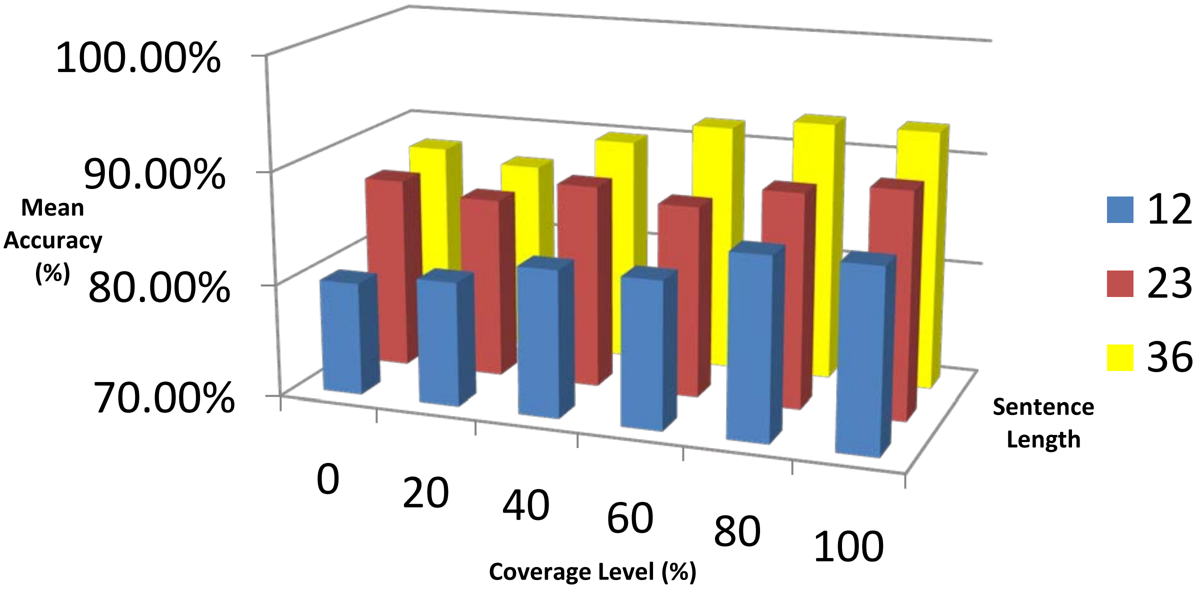


Figure 6 Impact of tag dictionaries on Accuracy

5.2. Descriptive Analysis

At the end of the user study participants had a chance to provide a free-text response regarding the user study as a whole. Most participants did not provide any feedback. However, those that did could be grouped into a few categories: questions regarding part-of-speech annotation in general, the study took too long to complete, those that did not like being forced to achieve high accuracy on the tutorial sentences, and those who felt it was a good exercise. These are broad categorizations and there were other responses; however, by far the most common were those who felt the study was too long and did admit that by the end they were less cautious with the annotation process and those that were unhappy with the high accuracy required on the tutorial sentences.

With this specific feedback in mind it is important to consider that in a real-life tagging scenario the annotators should only need to be trained on the system once. In addition, annotators would usually do sentences when they could and for the amount of time they desired, which should result in less fatigue while performing the annotation task. In turn, less fatigue while performing the annotation task should translate to faster times and higher accuracy.

5.3. Effects of Dictionary Size

A potential confounding factor for this study is the affect of the number of dictionary entries on the accuracy and time of token annotation. To determine the effects of the dictionary size a post-hoc analysis was done by grouping the annotated tokens according to the number of entries in each dictionary and then averaging the accuracy and annotation time for each group. Table 3 shows the number of tokens annotated in each dictionary size group along with the average accuracy and annotation time for tokens with the specified number of dictionary entries. With no dictionary the average accuracy is 83.92% and the average time is 8.85 seconds. One entry in the dictionary dramatically increases the accuracy to 95.84% and decreases the annotation time to 4.18 seconds, which is greater than 50% decrease. However, for dictionaries with more than one entry there is a downward trend in accuracy and an upward trend in time. Annotation of tokens with three entries is one exception to the trend but it is only a slight increase of accuracy and decrease of time. On the other hand, annotation of tokens with five dictionary entries does not fit the trend. There is a significant increase in annotation accuracy (92.94%) and a significant decrease in annotation time (4.34 seconds) which is comparable to the accuracy and time for the tokens with one dictionary entry. An analysis of the tokens that had five dictionary entries shows that the discrepancy is due to the tokens with five dictionary entries. There are only six tokens that have five dictionary entries and two of those tokens—"the" and "a"—are 67.47% of those tokens annotated. The skewed numbers could result from the fact that "the" and "a" are nearly always determiners. Unfortunately there is not sufficient data for tokens with more than two dictionary entries to provide a thorough analysis. These results imply that there is a negative correlation between both time and accuracy and dictionary size. On the other hand, it seems that there is nearly always an improvement of time and accuracy when the annotator has access to a dictionary of any size as compared to having no access to a dictionary.

Table 3 Accuracy and annotation time by dictionary size

Dictionary Size	Distinct Tokens	Tokens Annotated	Average Accuracy	Average Time (sec)
0	262	6994	83.92%	8.85
1	227	5330	95.84%	4.18
2	84	870	82.87%	7.38
3	41	475	86.74%	6.94
4	11	100	59.00%	9.79
5	6	255	92.94%	4.34
6	2	34	50.00%	13.45

6. Conclusion

Annotated corpora are being used more and more by computational linguists and computer scientists. However, creating annotated corpora is costly, which means researchers must either search for existing annotated corpora—which limits the number of possible sources—or they must find a way to reduce the cost of corpus annotation. There are myriad approaches to reducing the cost of POS annotation for a corpus including using software tools for annotation, statistical algorithms, or a POS tag dictionary. This project has shown that using a tag dictionary with significant coverage (in this project, 60%) during the annotation process improves both speed and performance of human annotators performing a part-of-speech (POS) annotation task.

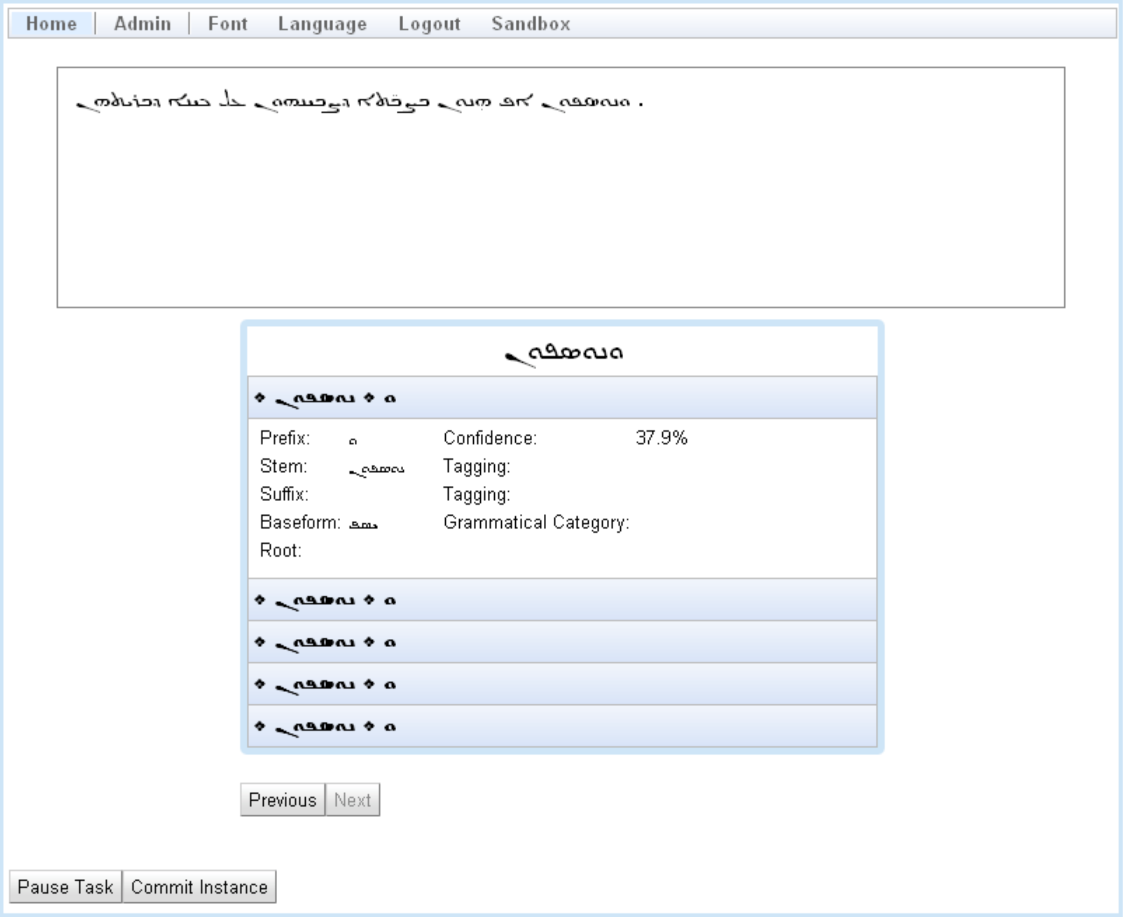
As is the case with any user study, these results are only viable given a specific set of criteria. First, these results are only valid for an English POS annotation task using the Penn Treebank tagset. Using a different tagset that is either more complex or simple could change the results. For example, if the tagset were as simple as what is taught in some preparatory schools English grammar classes (i.e., noun, verb, adjective, etc.) then this task could be considerably easier. In addition, the results of this study could change significantly using a morphologically complex language like Syriac. To emphasize the complexity of Syriac and the effect a language like this would have on the annotation task a sample annotation is provided in Figure 7. This one Syriac word contains as much information as the English phrase “to your king” and in an annotation task would first need to be segmented and then each meaningful segment would need one or more annotations. Moreover, the results may vary based on the subjects of the user study. The user study for this project was performed using first year graduate students in a linguistics program. These students had varying degrees of skill with regards to POS tagging as well as the

English language in general. However, despite all of these differences I believe that the general result would hold true even if the specifics were changed. Utilizing a dictionary of suggested answers would improve speed and accuracy of a human annotator after a certain level of dictionary coverage regardless of the specifics of the task.

Word:	LMLKKON	
Segmentation:	L (prefix) MLK (stem) KON (suffix)	
Definition:	to your (masculine plural) king	
Baseform:	MLK;	
Root:	MLK	
Stem Tagging:		
	Gender	Masculine
	Person	None
	Number	Singular
	State	Emphatic
	Tense	None
	Form	None
	Grammatical Category	Noun
Suffix Tagging	Gender	Masculine
	Person	Second
	Number	Plural

Figure 7 Example annotation for Syriac

7. Future Work



ܘܢܦܩܘܢ ܐܦ ܗܢܘܢ ܒܝ̈ܬܐ ܕܝܚܝܕܝܗܘܢ ܥܠ ܚܝܠܐ ܕܒܪܝܬܗܘܢ .

ܘܢܦܩܘܢ

ܘ ◆ ܢܦܩܘܢ ◆

Prefix: ܘ
Stem: ܢܦܩܘܢ
Suffix:
Baseform: ܢܦܩ
Root:

Confidence: 37.9%
Tagging:
Tagging:
Grammatical Category:

ܘ ◆ ܢܦܩܘܢ ◆

ܘ ◆ ܢܦܩܘܢ ◆

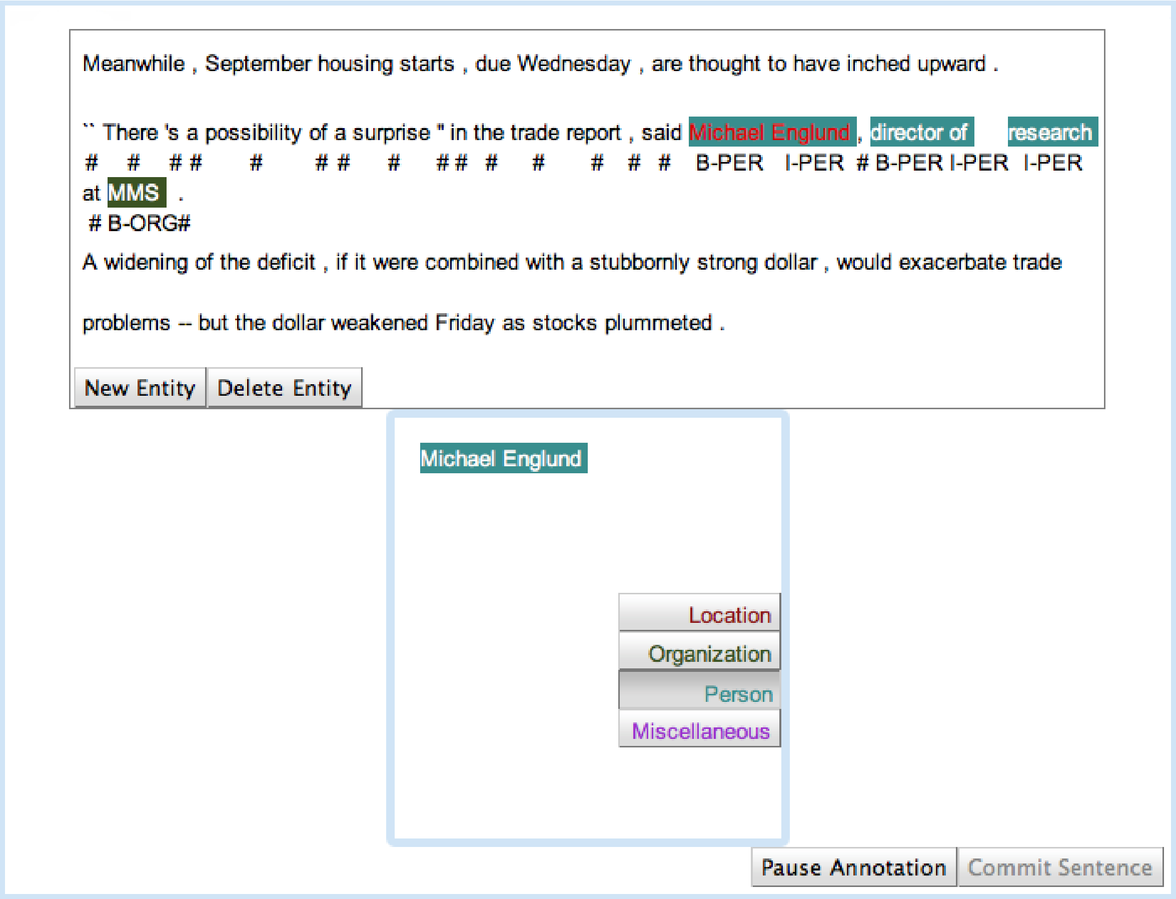
ܘ ◆ ܢܦܩܘܢ ◆

ܘ ◆ ܢܦܩܘܢ ◆

Figure 8 Syriac POS Annotation Interface

The morphological simplicity of English makes for an excellent pilot study because it reduces the complexity of the data analysis. On the other hand, it does little to support the extension of the hypothesis to more morphologically complex languages. As a result, other similar user studies must be completed using the same hypothesis: a part-of-speech tag dictionary will improve, in time and accuracy, the manual annotation of a text corpus. At the current time, the Brigham Young University (BYU) Natural Language Processing research group is working with the Center for Preservation of Ancient Religious Texts (CPART) to continue this direction of research and is in the process of implementing a similar user study using Syriac, a morphologically complex Semitic language. In fact, due to the extensibility of the CCASH project the development team has begun work on the user interface for the Syriac annotation project which can be seen in Figure 8.

In addition to testing the hypothesis in the context of texts written in other languages it is also important to determine the effect of a dictionary on other types of text annotation. Part-of-speech tagging is just one form of text annotation. For example, the Penn Treebank includes both POS tags and syntactic tags that show how a sentence would be parsed. In addition, researchers need to be able to annotate text at different levels and for different linguistics purposes. For example, a corpus demonstrating phonological phenomenon could be annotated for syllable boundaries and prosodic features (McEnery, Xiao, and Tono 2006). Morphological annotation would include a need for segmenting words into prefixes, suffixes, roots, and stems. Lexical annotation not only includes POS tags but lemma and semantic information. Syntactic analysis requires parsing similar to the Penn Treebank that shows the different syntactic levels. Additional types of annotation include, but are not limited to, coreference annotation, pragmatics, and stylistics. Each one of these different types of annotation could potentially require a different user interface. The CCASH framework currently has a user interface, which can be seen in Figure 9 below, available for annotating named entities which is used for named entity recognition (NER). The POS annotation user interface (both English and Syriac) and the NER user interface are just examples of what can be done using CCASH. We believe that if implemented correctly any of these user interfaces will help to reduce the cost of corpus annotation.



Meanwhile , September housing starts , due Wednesday , are thought to have inched upward .

`` There 's a possibility of a surprise " in the trade report , said Michael Englund , director of research
B-PER I-PER # B-PER I-PER I-PER
at MMS .
B-ORG#

A widening of the deficit , if it were combined with a stubbornly strong dollar , would exacerbate trade problems -- but the dollar weakened Friday as stocks plummeted .

Figure 9 NER interface for the CCASH framerwork

In chapter 2 I mentioned and provided information on a variety of existing tools as well as information regarding computer algorithms that can potentially speed up the annotation process. In chapter 3 I briefly discussed that one of the reasons for the technologies that we have chosen to use is that it will allow for the research team to integrate existing computational algorithms into the CCASH interface similar to many of the software packages already available. The addition of existing or even new algorithms could allow for an annotation process similar to the Penn Treebank which was first automatically annotated by a machine and then manually corrected by a team of researchers (Marcus, Santorini, and Marcinkiewicz 1994). The automate/correct paradigm is one that has merit and can be very useful. This is just one example of the cost decrease that the CCASH system can provide. However, this type of system requires already existing corpora to train the algorithm with. Using a semi-supervised learning algorithm reduces or removes the need for existing corpora.

7.1. Active Learning

As mentioned in chapter 2 active learning (AL) is a machine learning method that proposes using a considerably smaller subset of data to train the algorithm. Then using different techniques the algorithm determines which chunks of data will provide the most useful information for the task at hand and then an oracle, for example a human annotator, is asked to provide the required information by either annotating the text or correcting the machine annotation. Once that information is submitted the algorithm is retrained using that new piece of information. This process is continued until a specific goal is met. Using data from a user study with 47 annotators, Ringger, et al., (2008) were able to determine an hourly cost model for an English POS annotation task. Using the hourly cost model Haertel, et al., (2008) were able to measure the cost reduction using different AL algorithms. In summary, according to recent research using AL reduces the cost of an annotation task.

Simply using an AL algorithm can help to improve the speed of corpus annotation. However, this can also be applied to a dictionary implementation. It is common in most languages that a written word can server multiple morphological functions. For example, the noun address, as in a street address, and the verb address, to speak to, is spelled the same but perform different functions in a sentence. This is a fairly simple example that would be sorted out using sentence context. But there are many of these types of words that are not so simply separated. As a result, a dictionary would need to provide either all of the possible previous annotations or only the most likely. The AL algorithm comes into play after the oracle provides feedback on this word. The algorithm utilizes that input to adjust the algorithm and in the future it may not need the oracle to provide that information again or it simply may provide the human annotator with the correct answer. Obviously anytime the algorithm reduces the amount of time an annotator must spend on a word it is reducing the total amount of annotation time and cost.

Determining the affect of an active learning algorithm during an actual annotation task is a step that is very important for the future of this line of research. As a result, there are currently members of the CCASH development team that are working to incorporate an active learning algorithm into the CCASH environment. Conducting a follow-up user study using the same data and a group of participants with a similar profile would provide an interesting perspective on the affect of active learning on dictionary creation and the accuracy and speed of human annotators.

Where Does This Lead?

The goal of this project was to test one method of corpus annotation cost reduction. A slightly tangential question, although still relevant and related to this work, is how do you get started? CCASH provides a great starting point for any type of annotation project. It allows developers to integrate new and existing statistical algorithms with unique and specialized user interfaces for specific tasks. However, if a project is really being done from the ground up then there are several steps that must occur first. The research group must determine what has been done previously for the language and type of annotation as well as any existing resources that can be utilized. In addition, the annotation data set, which I have called a tag set in the case of part-of-speech annotation, must be determined and the exact task at hand must be defined. For example, Syriac part-of-speech annotation also includes segmentation and as a result the user interface must take that into account. Assuming some previous work has been done for the desired language and annotation type then the remainder of the process is simply determining which statistical algorithms to utilized and working out any kinks in the user interface for the corpus annotation. If, however, no previous work has been done then the research group and there is not a specialist available for the desired language and the specific task at hand then the team will begin from scratch.

Creating and annotating a corpus with no prior information presents a very interesting question. Is it possible to utilize existing statistical machine learning algorithms and user input to learn a model for the language? I propose that with sufficient effort put into data analysis and data input it would be possible to create system that begins with a clean slate and utilizes user input to determine a statistical model for the language. As user input is received and the statistical model is developed a dictionary would be created utilizing the annotations that are made. This is an area of future research that could provide very interesting results with regards to lesser-resourced languages.

7.2. Summary

Cost-reduction of annotating corpora has many fruitful paths of research and this is just a springboard for many of those. They include user interface and algorithm enhancements as well as analysis of different languages and different annotation tasks. The CCASH interface and this user study provide a base for future work to be completed and a framework to begin with.

8. References

Brants, Thorsten. 2000. TnT: a statistical part-of-speech tagger. *Proceedings of the sixth conference on applied natural language processing*, 224-231. Seattle, Washington: Association for Computational Linguistics.

Brill, Eric. 1992. A simple rule-based part of speech tagger. *HLT '91: proceedings of the workshop on speech and natural language*, 112-116. Harriman, New York: Association for Computational Linguistics.

Cunningham, Hamish, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: an architecture for development of robust HLT applications. *ACL '02: proceedings of the 40th annual meeting on Association for Computational Linguistics*, 168–175. Philadelphia, Pennsylvania: Association for Computational Linguistics.

Davies, Mark. 2009. The 385+ million word Corpus of Contemporary American English (1990–2008+): design, architecture, and linguistic insights. *International Journal of Corpus Linguistics* 14.159-190. doi:10.1075/ijcl.14.2.02dav.

Fort, Karën, and Benoît Sagot. 2010. Influence of pre-annotation on POS-tagged corpus development. *The fourth ACL linguistic annotation workshop*. Uppsala, Sweden: Association for Computational Linguistics.

Haertel, Robbie, Kevin Seppi, Eric Ringger, and James Carroll. 2008. Return on investment for active learning. *Proceedings of the NIPS workshop on cost-sensitive learning*. ACL Press.

Marcus, Mitchell P, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19.313–330.

McEnery, Tony, Richard Xiao, and Yukio Tono. 2006. *Corpus-based language studies: an advanced resource book*. New York: Routledge.

Menke, Joshua, and Tony R. Martinez. 2004. Using permutations instead of student's t distribution for p-values in paired-difference algorithm comparisons. *Proceedings of 2004 IEEE international joint conference on neural networks*, 2:1331-1335.

Morton, Thomas, and Jeremy LaCivita. 2003. Word-Freak: an open tool for linguistic annotation. *Proceedings of the 2003 conference of the North American chapter of the Association for Computational Linguistics on human language technology: demonstrations*, 17-18. Edmonton, Alberta, Canada.

Ogren, Philip. V. 2006. Knowtator: a Protégé plug-in for annotated corpus construction. *Proceedings of the 2006 conference of the North American chapter of the Association for Computational Linguistics on human language technology*, 273-275. New York.

Palmer, Alexis, Taesun Moon, and Jason Baldridge. 2009. Evaluating automation strategies in language documentation. *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, 36–44. Boulder, Colorado.

Ringger, Eric, Marc Carmen, Robbie Haertel, Kevin Seppi, Deryle Lonsdale, Peter McClanahan, James Carroll, and Noel Ellison. 2008. Assessing the costs of machine-assisted corpus annotation through a user study. *Proceedings of the sixth international language resources and evaluation (LREC'08)*. European Language Resources Association (ELRA).

Tomanek, Katrin, Joachim Wermter, and Udo Hahn. 2007. Efficient annotation with the jena annotation environment (JANE). *Proceedings of the Linguistic Annotation Workshop*, 9–16. Prague, Czech Republic: Association for Computational Linguistics.

Appendix A

Document provided to the user study participants.

URL: http://cash.cs.byu.edu/Ccash/EnglishUserStudy.html

Name: ____

Tag	Description	Examples
CC	Coordinating conjunction	and, or, both, either, neither
CD	Cardinal number	top, fifteen, 3
DT	Determiner	the, this, each, any, some, these, those
EX	Existential *there*	there
FW	Foreign word	de, en, ad hoc, en masse,
IN	Preposition or subordinating conjunction	in, of, although, when, that
JJ	Adjective	happy, bad, sixth, last, many
JJR	Adjective, comparative	happier, worse
JJS	Adjective, superlative	happiest, worst
LS	List item marker	1), 2), A., B.
MD	Modal	'll, can, could, might, may
NN	Noun, singular or mass	aircraft, data, woman, book
NNS	Noun, plural	women, books, Sundays, weekdays
NNP	Proper noun, singular	London, Michael
NNPS	Proper noun, plural	Australians, Methodists
PDT	Predeterminer	both, quite, all, half
POS	Possessive ending	's, '
PRP	Personal pronoun	I, me, you, he, them
PRP$	Possessive pronoun	my, your, mine, yours
RB	Adverb	very, so, to, enough, indeed, here, there, now
RBR	Adverb, comparative	further, gloomier, grander
RBS	Adverb, superlative	best, biggest, bluntest
RP	Particle	up, off, out
SYM	Symbol	Should be used for mathematical, scientific or technical symbols
TO	*to*	To
UH	Interjection	uh, well, yes, my
VB	Verb, base form	take, live, do, have, be
VBD	Verb, past tense	took, lived, did, had, were, was
VBG	Verb, gerund or present participle	taking, living, doing, having, being
VBN	Verb, past participle	taken, lived, done, had, been
VBP	Verb, non-3rd person singular present	take, live, do, does, am, 'm, are, 're
VBZ	Verb, 3rd person singular present	takes, lives, does, has, been
WDT	Wh-determiner	which, whatever, "that" when it is used as a relative

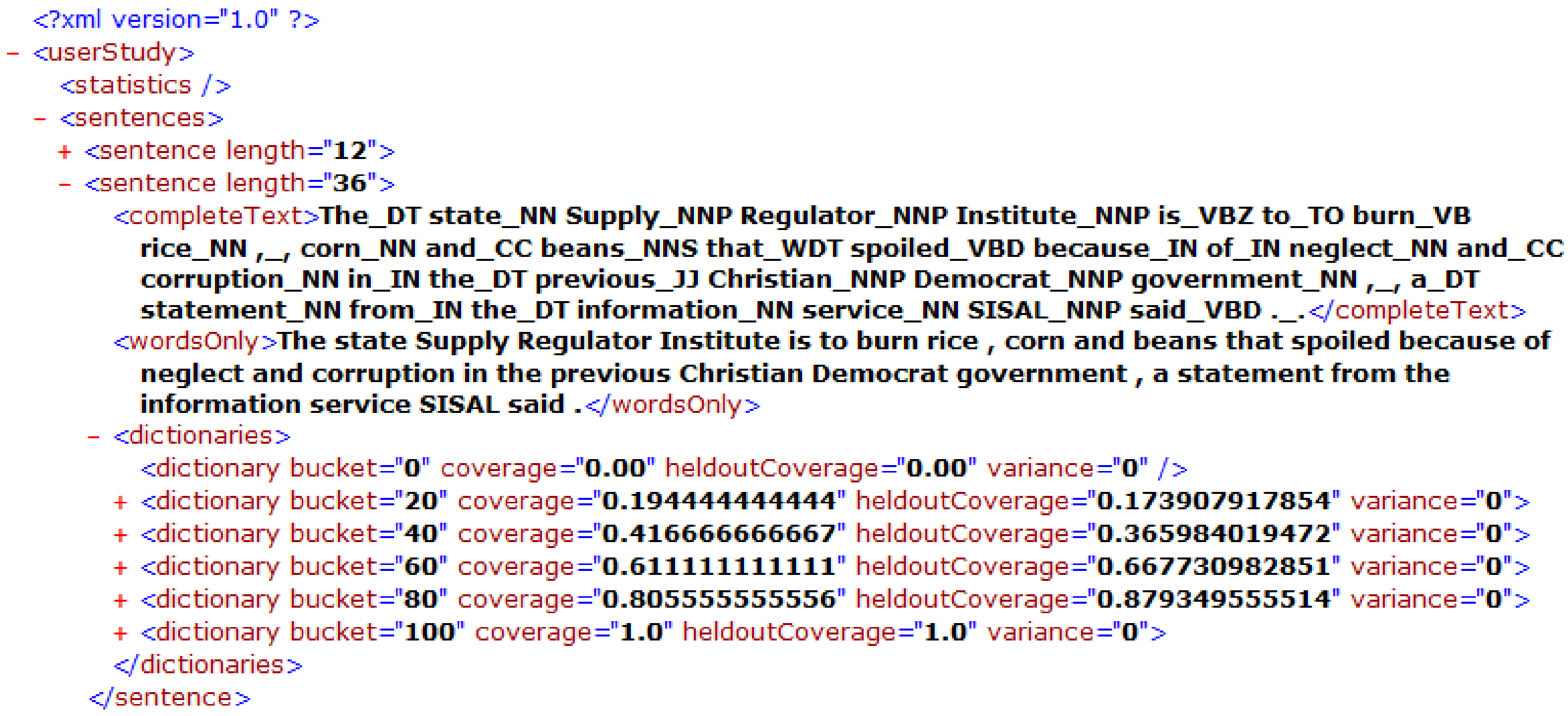
		pronoun
WP	Wh-pronoun	who, whoever
WP$	Possessive wh-pronoun	Whose
WRB	Wh-adverb	when, how, why, however
-LRB-	Left Curly Brace/Parentheses	{ , (
-RRB-	Right Curly Brace/Parentheses	} ,)
.	Sentence Final Punctuation	., ?, !
:	Colon, Semi-Colon, M-Dash	:, ;, --
,	Comma	,
-	Dash	-
$	Monetary Units	$
``	Opening Quotation Mark	``
"	Closing Quotation Mark	"
#	Pound Symbol	#

If you run into any major problems that prevent you from completing the user study then please send an email to ccashstudy@gmail.com with as much detail as possible.

Serial Number:

Appendix B

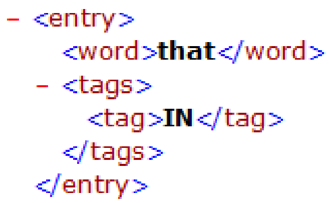
Due to the length of the XML file containing the sentences and all of the dictionaries for a sentence the XML file itself is not included in this report. Below are a collection of screenshots that demonstrate the overall format of the XML file as well as a sample



```
  <?xml version="1.0" ?>
- <userStudy>
    <statistics />
  - <sentences>
    + <sentence length="12">
    - <sentence length="36">
        <completeText>The_DT state_NN Supply_NNP Regulator_NNP Institute_NNP is_VBZ to_TO burn_VB
          rice_NN ,_, corn_NN and_CC beans_NNS that_WDT spoiled_VBD because_IN of_IN neglect_NN and_CC
          corruption_NN in_IN the_DT previous_JJ Christian_NNP Democrat_NNP government_NN ,_, a_DT
          statement_NN from_IN the_DT information_NN service_NN SISAL_NNP said_VBD ._.</completeText>
        <wordsOnly>The state Supply Regulator Institute is to burn rice , corn and beans that spoiled because of
          neglect and corruption in the previous Christian Democrat government , a statement from the
          information service SISAL said .</wordsOnly>
      - <dictionaries>
          <dictionary bucket="0" coverage="0.00" heldoutCoverage="0.00" variance="0" />
        + <dictionary bucket="20" coverage="0.194444444444" heldoutCoverage="0.173907917854" variance="0">
        + <dictionary bucket="40" coverage="0.416666666667" heldoutCoverage="0.365984019472" variance="0">
        + <dictionary bucket="60" coverage="0.611111111111" heldoutCoverage="0.667730982851" variance="0">
        + <dictionary bucket="80" coverage="0.805555555556" heldoutCoverage="0.879349555514" variance="0">
        + <dictionary bucket="100" coverage="1.0" heldoutCoverage="1.0" variance="0">
        </dictionaries>
      </sentence>
```

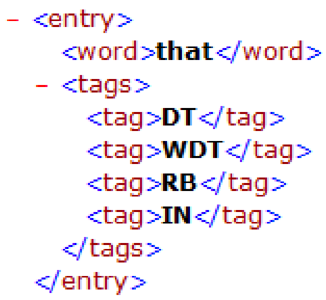
Figure 10 General XML format used for the dictionaries

The XML includes the entire sentence, with and without tags, as well as a list of dictionary entries at each coverage level. The actual coverage level and the coverage level against the held-out data are stored in the XML as well.



```
- <entry>
    <word>that</word>
  - <tags>
      <tag>IN</tag>
    </tags>
  </entry>
```

Figure 12 Tags for “that” in the 20% coverage dictionary



```
- <entry>
    <word>that</word>
  - <tags>
      <tag>DT</tag>
      <tag>WDT</tag>
      <tag>RB</tag>
      <tag>IN</tag>
    </tags>
  </entry>
```

Figure 13 Tags for “that” in the 80% coverage dictionary

entry for the word “that” in dictionaries with coverage levels of 20% and 80%.

These two sample entries are for the same word in the same sentence. Because the dictionaries are built using the Penn Treebank corpus, all of the tags listed in Figure 13 occur at least once in the corpus for the word “that”. This demonstrates the possible confusion that could occur with dictionaries with high levels of coverage. On the other hand, Figure 11 provides the correct answers for the sentence that contains these two dictionary entries and in this case the one tag provided by the dictionary with 20% coverage is incorrect but the dictionary with 80% coverage does contain the correct tag.