MS Thesis Approval

THESIS TITLE: Analysis of Named Entity Recognition & Entity Linking in Historical Text

AUTHOR: Karim Assads

M.S. Thesis Approved as an elective towards the Master of Science Degree in Computer Science.

Thesis Advisor: [Signature]
Date: 8/25/2016

Thesis Reader: [Signature]
Date: 8/12/2016

Thesis Reader: [Signature]
Date: 

Thesis Reader: [Signature]
Date: 

Thesis Reader: [Signature]
Date: 

GRADUATE SCHOOL APPROVAL:

Director, Graduate School: [Signature]
Date: 8/31/2016

COPY RECEIVED IN GRADUATE SCHOOL OFFICE:

Recipient's Signature: [Signature]
Date: 9/6/2016

Distribution: Once completed, this form should be scanned and attached to the front of the electronic dissertation document (page 11). An electronic version of the document can then be uploaded to the Northeastern University-UMI website.
Analysis of Named Entity Recognition & Entity Linking in Historical Text

Thesis
By Kunal Asarsa
Abstract

Analysis of Named Entity Recognition & Entity Linking in Historical Text

Author: Kunal Asarsa
Advisor: David A Smith

Named Entity Recognition is a Natural Language Processing (NLP) technique that is used to identify names of people, places, organizations, and more from a given piece of text. Entity Linking is the additional NLP task that involves connecting a machine identified Named Entity to a Knowledge Base Entity.

Both these techniques have had fair share of success with more recent content, where for example names have been linked to entities on Wikipedia (a process called "wikification"). However, parsers, models and other NLP tools tend to act a little differently with historical text. As per our initial research, there are often issues like ‘names that do not exist in models’ or ‘words that are no longer used’, ‘words that are now spelled in a different way’ and more. This study aims to minimize the effect of such differences by modifying the NLP processes and then comparing the manual tagging of a sample corpus versus the tagging and linking performed by the machine.

With this study, we aim to arrive at changes required in NLP tools and/or inclusion of additional steps to better handle historical text and compare this to the performance of the tools without modification. We hope that this study cannot only provide statistical results but also findings that can potentially help expedite & improve the process of handling historical text.
## CONTENTS

List of Tables .................................................................................................................. vi
List of Figures .................................................................................................................. vii
Abbreviations .................................................................................................................. viii

**Chapter 1** Introduction ................................................................................................. 1
  1.1 Current challenge ................................................................................................. 1
    1.1.0 Historical text – Generic challenges ......................................................... 2
    1.1.1 Corpus Specific challenges ......................................................................... 2
  1.2 Goals and Milestones ......................................................................................... 3

**Chapter 2** Related Work ............................................................................................. 4
  2.1 Historical Text ..................................................................................................... 4
  2.2 Challenges with historical text ........................................................................... 4
  2.3 Text Extraction .................................................................................................. 5
  2.4 Other Analysis .................................................................................................... 5
  2.5 Novel characteristic of current work ................................................................. 6

**Chapter 3** Corpus ....................................................................................................... 7
  3.1 Information about corpus ................................................................................... 7
  3.2 Why this corpus and what makes it a good study option ................................... 8

**Chapter 4** Stages of Experimentation ....................................................................... 9
  4.1 K B Bridge .......................................................................................................... 9
  4.2 Stanford CoreNLP ............................................................................................. 9
  4.3 Berkeley entity ................................................................................................... 10
    4.3.0 Preprocessing ............................................................................................... 10
    4.3.1 Tweaking system .......................................................................................... 11
    4.3.2 Using the system ......................................................................................... 11
  4.4 Enhancements to Text Extraction ...................................................................... 12
    4.4.0 Stages and changes ...................................................................................... 12
  4.5 Post-processing .................................................................................................. 12
    4.5.0 Tag Correction .............................................................................................. 13

**Chapter 5** Results ...................................................................................................... 14
  5.1 Stanford ............................................................................................................... 14
    5.1.0 Result File Structure .................................................................................... 14
LIST OF TABLES

Table 4.1 Preprocessing output
Table 4.2 Berkeley NER output
Table 4.3 Statistics for single token vs multi-token NER
LIST OF FIGURES

Figure 5.1 Stanford NER stats with stage I Text Extraction .................................................. 16
Figure 5.2 Stanford NER stats with stage II Text Extraction ............................................... 17
Figure 5.3 Graph showing Recall percentages for NED on corpus documents ....................... 21
Figure 5.4 Single token Vs. Multi Token Recall value comparison ..................................... 24
Figure 5.5 Single token Vs. Multi Token Precision value comparison .................................. 25
Figure 5.6 Histogram for Ratio of tokens from Single token words vs. multi token words ...... 25
Figure 5.7 Recall value comparison ..................................................................................... 26
Figure 5.8 Precision value comparison .................................................................................. 26
Figure 5.9 Scatter plot of proportion values for adams.jews.xml ....................................... 27
Figure 5.10 Recall Values Before and After Tag Correction .............................................. 29
Figure 5.11 Precision Values Before and After Tag Correction ........................................... 29
Figure 6.1 Variance in Recall .............................................................................................. 32
Figure 6.2 Variance in Precision .......................................................................................... 33
Figure 6.3 Recall for NER with diff Text Extraction techniques ........................................... 33
Figure 6.4 Change in Recall ............................................................................................... 34
Figure 6.5 Precision for NER using diff Text Extraction techniques .................................... 34
Figure 6.6 Change in Precision .......................................................................................... 35
Figure 6.7 Recall Change due to Post Processing ................................................................. 36
Figure 6.8 Recall change percentage difference Histogram ............................................... 36
Figure 6.9 Precision Change due to Post Processing ............................................................. 37
Figure 6.10 Precision change percentage difference Histogram ........................................... 37
## ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NER</td>
<td>Named Entity Recognition</td>
</tr>
<tr>
<td>NEL</td>
<td>Named Entity Linking</td>
</tr>
<tr>
<td>NED</td>
<td>Named Entity Disambiguation</td>
</tr>
<tr>
<td>WWP</td>
<td>Women Writer Project</td>
</tr>
<tr>
<td>WWO</td>
<td>Women Writer Online</td>
</tr>
<tr>
<td>POS</td>
<td>Part of speech</td>
</tr>
<tr>
<td>CONLL</td>
<td>Conference on Natural Language Learning</td>
</tr>
<tr>
<td>KBB</td>
<td>Knowledge Base Bridge</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>ST</td>
<td>Single Token</td>
</tr>
<tr>
<td>MT</td>
<td>Multi Token</td>
</tr>
</tbody>
</table>
Chapter 1  INTRODUCTION

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

Named entity linking (NEL) is a further NLP task that involves connecting a machine identified named entity to a knowledge base entity. The Entity Linking process also involves of a finishing step, where selection is made between two or more contender knowledge base entities (that are very similar and create ambiguity). This is often referred to as Named Entity Disambiguation.

Both these techniques have had fair share of success with more recent content, where for example names have been linked to entities on Wikipedia (a process known as “wikification”). But parsers, models and other NLP tools tend to act a little different with Historical text. As per our initial research, there are often issues like ‘names that do not exist in models’ or ‘words that are no longer used’, ‘words that are now spelled in a different way’ and more. This study aims to minimize the effect of such differences by modifying the NER techniques and then comparing the manual tagging of a sample corpus versus the tagging and linking performed by the machine.

With this study, we aim to arrive upon changes required in NLP tools to better handle historical text and compare this to the performance of the tools without modification and on different corpus that would act as baseline.

1.1  CURRENT CHALLENGE

For this study, we worked with WWW (corpus), which was a full-text collection of early women’s writing in English, published by the Women Writers Project at Northeastern University. It includes full transcriptions of texts published between 1526 and 1850, focusing on materials that are rare or inaccessible. The range of genres and topics covered makes it a truly remarkable resource for teaching and research, providing an unparalleled view of women’s literate culture in the early modern period.

This corpus provides quite a few challenges when it comes to performing NLP techniques. Below you will find some of the challenges that were a focus of this study.
1.1.0  Historical text – Generic challenges
The first and foremost challenge was to work with text that is different from current newspaper text in numerous ways. As most of the widely available parsers and NLP toolkits are trained and tested with modern text, historical text becomes a challenge while performing NLP techniques on it.

To better explain these differences, let us look at them:

- **Language** – considering that the text belongs to era of 16th to 19th century, the most striking feature that can be observed is the language itself. English has evolved over time and it is evident that there are differences in the language, for example, when you compare the text in today’s newspaper with the works of Shakespeare.
- **Names** – Just like the language, the names of people, places and organizations have evolved over time. The models used today to train the NLP tools, are mostly comprised of names collected from sources that are more recent, thus falling short on accurate identification of named entities (not complete failure).
- **Classic Latin alphabets** – the classic Latin alphabets, that were widely used till the middle ages did not comprise of the letters J, U, W. This is observable in the current corpus where tokens have I, V replacing the J, U in currently known spellings. Example Institute vs institvte.

1.1.1  Corpus Specific challenges
Considering the challenges described in the aforementioned section, it becomes important that we pre-process the text before using an NLP toolkit, being used as a black box (no modification to inner mechanisms). In our study, we have Java code (independent from the NLP tools used), that takes text files from corpus and converts it into plain text. In addition to handling the challenges due that arise due to historical text, there are few more challenges related to the input text format.

- **Format** – the documents within corpus are books that exist as XML documents. XML documents are not something that the NLP tools (used in study) as designed to work with. Additionally, they not have a very specific structure that cannot be directly converted to Plain text. Thus in our study we look at various pre-processing techniques that help extract text from XML documents.
- **Custom XML Tags** – An alternative to creating very own pre-processing scripts, was to use existing XML to text tools. However, the tools explored during the study were created to handle very standard set of XML tags, and not an extension of it. The corpus has XML tags that are unique to the corpus/or similar
documents, yet not universal. This means we had to handle them in order to obtain text.

- Tokenization / Breaking sentence – Sentence breaks were in general one of the biggest challenges faced during pre-processing stages. Removal of XML tags to get text, or conversely exclusion of tags to obtain tags, often led to missing or additional breaks in the sentences, which was bound to throw off the tools.

1.2 Goals and Milestones

After the short analysis of the challenges provided by the current corpus, in this section we discuss our goals for the study, as we try to incorporate the expectations and milestones for the study.

- Explore and validate the challenges faced while processing Historical text.
- Create Java code to ‘Extract text’ (Preprocess it).
- Study the effects of text extraction and benefits of preprocessing.
- Statistically show how improvements in text extraction can help solve these challenges.
- Generate NER results for the corpus using 3-class model and existing systems (Stanford CoreNLP, Berkeley Entity).
- Create Java code for analysis of NER output i.e. creating statistics for NER on each class, NER single-token vs multi-token, NER using different text extraction techniques.
- Based on analysis, deduce possible methods for post-processing the NER output in order to boost performance.
- Generate Java code for post processing and statistics for verifying the effect of post-processing.
- Run Berkeley Entity to produce Entity Linking outcome. Observe and analyze the outcome.
- Identify the challenges solved and propose future work/improvements to system.
Chapter 2 RELATED WORK

The different experiments carried out as a part of this study and the analysis on them, deal with certain categories or fields like NER, historical text etc. In this section, we look at following fields and the related (supporting and varying) work done in these fields by other researchers.

2.1 HISTORICAL TEXT

The rising interest in historical text can be accredited to the developments in digitization of historical text. This has been deeply explored and explained in various research papers like Crane et al., 2006 [10] and Borin et al., 2007 [11].

These papers (and many others) have shown how the increased availability of historical text, due to digitization of libraries collections and newspaper text, has sparked an interest in exploring the vast untapped knowledge that exists in these documents.

Ranging from cultural use to exploration of Civil War, historical text has had varying application and thus caught the attention of libraries and other organizations leading to various projects like Project Perseus [23], NORA and ARMADILLO [24].

2.2 CHALLENGES WITH HISTORICAL TEXT

While a lot of projects and research deals with historical text, only a few of them discuss the generic or specific challenges faced while working with historical text. Buchler et al., 2008 [16] discusses a few of these, like the use of special symbol for ‘s’ in historical text that gets transformed to ‘f’ due to its appearance and talk about the conversion to utf-8. Additionally they even mention other problems such as preserving the white spaces and new lines while turning data into plain text.

Another research that matches our work largely is Borin et al. 2007 [11]. This paper is about the analysis of NER on historical text and does cover generic challenges like spelling variations and unknown names. They even talk about structure preservation during transformation of the text for input to NLP tools. Although the corpus here is a collection of work of Swedish authors from 18th to 19th century, the problems faced are very similar to ones faced during our work and can be justly termed as generic.
2.3 Text Extraction
Consider the equivocal support of challenges faced during processing historical text, some steps are commonly identified as ones that can be handled during the preprocessing. Galibert et al. [16] talk about ‘block selection’ i.e., selecting blocks of text from newspaper text. They emphasize that not all text is important and unwanted text like headings or tables can throw off results. They even talk about ‘attribute correction’ as way of working with hyphenation and line breaks. However, there are no numbers statistically specifying the effect of these on final NER outcome.

2.4 Other Analysis
Here is a glance at other related work with notable points of study:

- Mikheev et al. [17] compares the results of not using a gazetteer vs using one with exhaustive list of names vs. using one with selective names. This study concluded that best way to use a gazetteer was to use one with selective list of names that have high probability of appearing in the corpus. In addition, the effect of gazetteer is arguable and affects the results heterogeneously i.e. at different extent for different classes.

- Stevenson et al. [22] have shown similar work with discussion on use of name-list generated from corpus for NER training. Additionally they made use of a technique similar to (but not same as) our post processing and showed an increase in recall but not in precision. The technique involved altering NER system to work with corpus generated list, thus tagging any occurrence of name that matched list. They also stated encountering low precision values for organizations class as the words within multi-token organizations also occurred independently as non-names.

- Gu et al. [21] & Bryne, 2007 [19] have done interesting work on nested NER. Although this is not directly comparable to our work: Gu et al. talks about discovering DNA terms and there is lower stress on corpus being historical text. They still provide interesting observations such as analysis of single token words vs. multi token words. The results from Bryne, 2007 differ from ours and show lower variations between single-token and multi-token words, which could be attributed to different ways of generating statistics and use of different NER systems.

- Aramaki et al., 2006 [12] have worked on de-identification of PHI which in turn works like NER. They have used ‘Token level majority’ & ‘Label consistency’ that work just like our ‘Majority class’ & ‘Tag correction’ technique; it shows
improvement in NER precision from 96.66% to 98.29%. With difference in corpus, tools, objective, etc. the results are not comparable but the use of technique is similar to post processing in our study. Primary difference here lies in the use of the method for NER in their paper vs. using it post NER in our study.

2.5 NOVEL CHARACTERISTIC OF CURRENT WORK

With the previous subsections, we got a chance to compare our work with the work done by researchers in similar field, looking at similarities and differences. This subsection points out the characteristics of current study that make it unique:

- A detailed exploration of challenges faced while working with historical text along with evidence showing how proposed solutions can help overcome them.
- Concentration on use of preprocessing & post processing techniques in order to adapt a generic system (no modification to the tool) to work with a very specific themed corpus.
- Comparative study of two existing NER systems (Berkeley Entity & Stanford CoreNLP).
- Creation and analysis of text extraction techniques; validating their importance in solving challenges faced while working with NER.
Chapter 3  

CORPUS

3.1 INFORMATION ABOUT CORPUS
The Women Writers Project is a long-term research project devoted to early modern women's writing and electronic text encoding. Our goal is to bring texts by pre-Victorian women writers out of the archive and make them accessible to a wide audience of teachers, students, scholars, and the general reader. The Women Writers Project maintains "Women Writers Online" an electronic collection of rare or difficult to obtain works written or co-authored by women from the sixteenth century to the mid nineteenth century. In addition, the WWP is actively engaged in researching the complex issues involved in representing manuscripts and early printed texts in digital form and holds an occasional conference, Women in the Archives, as well as teaching workshops in text encoding and other practices central to digital humanities.

The focus of the corpus is currently on hard to find or generally inaccessible texts from both well-known and more obscure writers. Authors included in the corpus include, among many others:

- Hannah Adams
- Anna Laetitia Barbauld
- Aphra Behn
- Margaret Cavendish
- Queen Elizabeth I
- Margaret Fell
- Felicia Hemans
- Catherine Parr
- Mary Sidney
- Mary Wollstonecraft
3.2 WHY THIS CORPUS AND WHAT MAKES IT A GOOD STUDY OPTION

With the short introduction about the corpus, the next question that arises is why this corpus? The WWP is a well-researched and academically used corpus. Below are a few reasons that make it a good fit for NER/NEL testing.

- The corpus is easily available and part of ongoing research at Northeastern University
- It is a live project and currently has people involved not only in its development/maintenance but also quite a few people who are working on it as researchers. It is also a part of course structure, allowing the addition of students to the list of people involved.
- The text collection is not generic collection of books but rather consists of books by ‘pre-Victorian women writers’.
- The corpus, belonging to century past, contains language that is a little different from English that we currently use.
- Another, not so unique yet important, characteristic is that the documents are manually tagged for Named Entities, thus making it possible to easily create statistics for the performance of NER systems used.

To summarize, these characteristics make the corpus unlike the generic text used to train and test the NER systems. This leads to a challenge of getting appreciable results with a corpus on which the system is likely to fail or perform badly. Thus providing an opportunity to study where the systems fail and what can be done to better them.

The samples of text from the corpus can be found in Appendix B.
Chapter 4  STAGES OF EXPERIMENTATION

4.1  K B BRIDGE
Knowledge Base Bridge - Entity linking system is a state of the art entity linking system used by UMass in the TAC KBP English entity link task. It uses a search index of Wikipedia and a supervised learning to rank system built upon the RankLib package [1].

At the very beginning of the study, K B Bridge was identified as a potential Entity Linking tool to be used for processing documents from WWO corpus. KBB was a relative new tool and had gone through some undocumented development. This led to some solicitation as to acquiring the correct version and set of instructions to run it. After multiple attempts at getting the system up and running and investing couple of weeks, it was determined that an alternative tool might be more suitable for the task.

Overall, KBB proved to be a stepping stone and not an active fragment of this study.

4.2  STANFORD CORENLP
Stanford CoreNLP is an integrated framework. Its goal is to make it very easy to apply a bunch of linguistic analysis tools to a piece of text. A CoreNLP tool pipeline can be run on a piece of plain text with just two lines of code. It is designed to be highly flexible and extensible [2].

Stanford CoreNLP is primarily used for its NER tool. With a large number of people from academia and research using this system, it was a great system to study and compare. Although not the prime focus of study at any given point, Stanford (known for its accuracy and speed) provides a good alternate solution for comparing results. In this case, Berkeley Entity (discussed later) was the prime option.

As mentioned earlier, Stanford CoreNLP system was used as a black box. Not only did Stanford provide quick results for comparison, but it also helped compare how each system handled the text and compare the effects of each tweak, individually on the text.
4.3 BERKELEY ENTITY

Berkeley Entity is a project from the Berkeley NLP group created by Greg Durrett, that jointly solves the problems of named entity recognition [3], coreference resolution [4], and entity linking with a feature-rich discriminative model.

Incorporating the existing Berkeley parser and Berkeley coreference resolution system, Berkeley Entity offers a complete solution towards extracting Named Entities and linking them to their Wikipedia entries. Thus covering three tasks aimed by this study i.e. NER, coreference, NEL.

The system is more research oriented tool than an end-to-end system, in the sense that it has great results for CONLL datasets (CONLL format), but show the same proficiency with plain text or any other format of data for that matter. Thus the initial work with Berkeley Entity system was to either use it with corpus being converted to CONLL format or the Corpus being converted to plain text (which the system can work with).

As the initial attempts to obtain CONLL datasets [5] (to tune system and verify results published earlier) failed, we moved to attempting conversion of WWO corpus to CONLL. This again proved a challenge as most tools that worked with CONLL data (or for CONLL challenge) depended on the task set released by the organization that made it grueling task to convert the data to CONLL format.

Without further spending time on the trivial task, it was decided to convert the corpus text (originally XML data) into plain text.

4.3.0 Preprocessing

With the Stanford and Berkeley systems setup and ready to use, they were initially tried with original XML documents to test them. The results were way off the expectations due to certain factors:

- Structured text – the systems are designed to work with plain text and were unable to complete all processes correctly, including sentence separation. This propagated the errors onto the next steps such as POS tagging and finally NER
- Existing Tags – the XML documents had tags for not only the Named Entity identifier but also other generic XML tags

Basically these complexities disallowed correct execution of the text. Hence a small preprocessing engine was needed before the documents could be readied for NER.

The initial approach was to get rid of tags and restructure the document in order to convert the XML docs into plain text. Alternatively, a few XML-to-text conversion tool/software were considered but most of them were not suited to the task/corpus
style. Thus a java file was created to generate plain text input for the systems. These files are named as base files and only consist of NER tags that we use for generating stats. The same code also generates a separate set of files, with these tags removed. These ‘No Tags’ files are used as input to the two systems.

4.3.1 Tweaking system
The Berkeley system allows for a certain set of parameters to be set in-order to tune the performance of the Berkeley Entity system for Entity Linking. Here is a list of tunings tried for this study:

- Wikifier
- NER features
- Preprocessing features

With the different settings in place, the system was tested upon a subset of the corpus to determine the effects on the output. To summarize, the changes had little to no effect on the outcome. The outcome (discussed later) have been impracticable and the tweaks did not change them enough to be considered workable.

4.3.2 Using the system
Finally, the Entity Linking results were deemed unviable and in need of work that would require more time and work hours, than permissible for a Master’s thesis. The tool was however not completely inadequate, as it did generate NER results that we could work with and with time improve.

For a glimpse at how the system was run, please refer to Appendix A.

The sample script shows commands used for running the Berkeley Entity system. As can be seen, the system first starts with the commands to preprocess the text. This is internal to the tool and should not be confused with the preprocessing done to extract text from XML documents. The preprocessing turns the text into format that can later be utilized by other tools in pipeline like the NER tool. It runs sentence splitter, tokenizer, POS tagger or a selection of these.

The preprocessing turns the plain text into an array/list of tokens that have been POS tagged. The command seen above is for joint system or NER + coreference + Wikification. Alternatively, Berkeley provides an option to run these tools individually. This study had initially started with the use of the joint system to obtain the Entity Linking results, but after impractical results, switched to NER only.
4.4  ENHANCEMENTS TO TEXT EXTRACTION
Aforementioned systems explain a little about how pre-processing plays an important part in the use of NER systems. Unlike the numerous formats or document types that the NER systems excel at processing, plain text seems to be the one compatible with most of them. So when we try to generalize the use of these systems, or conversely use the system with a documents that it wasn’t intended to be used, we require simple pre-processing to convert the documents into something that is system understandable. This leads to the need for a simple pre-processing part, that turns our corpus with xml documents and historical language into something that both Stanford and Berkeley systems can easily work with: Plain Text.

4.4.0  Stages and changes
The preprocessing for current study was done using JAVA. The scripts went through a long list of small changes to generate plain text that was resemble original render of XML as closely and accurately as possible while avoiding symbols, structure and language that could adversely affect NER outcome. These changes can be grouped into two main stages for the pre-processing

1.  Stage I
    This stage basically worked around using the original XML structure and then removing each component (tags, symbols, spaces, whitespace etc.) that wasn’t required.

2.  Stage II
    This stage basically worked around starting with what was inside the <Text> or <Body> tags (i.e. core content of the books) and then processing this first. Then adding other parts of the books that might be good to include for NER.

Overall, the stage II pre-processing easily outperformed stage I script and showed significant improvement in NER results. Also stage II made it easier to compare the results with base file to generate stats.

4.5  POST‐PROCESSING
With text‐extraction and the two systems now tuned and running, the last stage of the study aimed at post fixes, i.e. changes to the outcomes in order to improve NER results. After considering a few options as to how this could be done, we decided to implement a few and test them.
4.5.0 Tag Correction
Looking at the results, one of the most interesting observation was that a word that occurred n times through the document was often tagged m times only, where m < n. This begged to ask the question that whether the word should have been tagged the remaining (n-m) times.

- Intra document tag correction
  To start with, we wanted to look at the occurrences of a word within a document. So the part of the experiment was to make an assumption that the word should have been tagged for n-m occurrences too, but only for qualifying words. As a general observation, if m was closer to n, it showed us higher confidence in tagging all occurrences. Thus by using a belief number \( f = 0.75 \), we performed tag correction and as expected the results showed improvement.

- Cross document tag correction
  The same experiment can be extended to work with occurrences of each word throughout the corpus. The intra-document tag correction discussed above required 4 passes of each document, requiring substantial time for processing each document. After initial estimates, it was determined that with the available resources (time/memory) it would not be possible to successfully complete the cross document tag correction. But it does remain an interesting proposition that remains open to exploration in the future.
Chapter 5  RESULTS

5.1  STANFORD
The experiments during the study comprise two major sets of results. Each set is the result of same NER system (Stanford NER) without any changes to the system. The only difference is creation of the two sets is the way the text is initially extracted from the original XML documents in the corpus.

5.1.0  Result File Structure
Before we look at the output from each pass of NER system on the corpus, here is a brief look at the structure of ‘NER stats file’ that contains statistics for NER system on each document from corpus.

- Base File Stats
  This section of the file contains the simple statistics for the base file or the original text that has been preprocessed and converted to plaintext. It records total counts for ‘Total Tags’, ‘Person’, ‘Places’, ‘Organizations’, ‘Other Names’ within a file.

- Output File Stats
  This section of the file has statistics similar to previous section, but for the output file generated by the NER system

- Comparative Analysis (Strict)
  Comparative analysis is a way to generate NER statistics by comparing the output to base file. It looks at the files word by word and compares tags to determine match as True Positive, True Negative, False Positive or False Negative. These counts are then used to generate 3 values most frequently used to measure performance of NLP systems, i.e. Precision / Recall / f-measure. The three values are measured not only for all words from document (overall) but also for each class, i.e. Person, Place, Organization. Thus resulting in 12 (3values * 4classes) figures within the comparative analysis section.
Comparative Analysis (Permissive)
The second comparative analysis section tries to record the same 12 figures as mentioned in the previous section, with the difference being in its way of judging tokens as TP / TN / FP / FN.
The previous section does a word to word match, treating single token phrases and multi-token phrases alike. This section works a little different by considering a partial match (e.g. if 2 out of 3 tokens from a multi-token phrase are tagged correctly) as complete match. This is a more lenient way of assessing the system.
Chapter 5 Results

5.1.1 Preliminary results
The preliminary results were generated using stage 1 preprocessing technique as explained in section 3.4.1. This method was original choice for extracting text and consisted of sentences from all over the books i.e. headings and notes included (later identified as possible cause for lower accuracy of NER system).

Sample Output File:

Base File Stats:
Total Tags: 6301.0
Person Names Recorded: 3493.0
Places Recorded: 2727.0
Organizations Recorded: 81.0
Other Names Recorded: 0.0

Output File Stats:
Total Tags: 6308.0
Person Names Recorded: 3668.0
Places Recorded: 2437.0
Organizations Recorded: 203.0

Comparative Analysis (Strict):
Overall Recall: 0.6973496270433265
Person Name Recall: 0.61580303464071
places Recall: 0.8217821782178217
Organizations Recall: 0.024691358024691357
Overall Precision: 0.7959733671528219
Person Name Precision: 0.7977099236641222
places Precision: 0.853508411981945
Organizations Precision: 0.07389162561576355
Overall f-measure: 0.743404786342602
Person Name f-measure: 0.6950515577083779
places f-measure: 0.8373448836339302
Organizations f-measure: 0.03701418877236274

Comparative Analysis (Permissive):
Overall Recall: 0.8203777178225681
Person Name Recall: 0.8145052008779468
places Recall: 0.8448848844884484
Organizations Recall: 0.248559670781893
Overall Precision: 0.7959733671528219
Person Name Precision: 0.7977099236641222

Figure 5.1 Stanford NER stats with stage I Text Extraction
5.1.2 Final Results
The final results are outcome of a slightly different text extraction as compared to the stage 1 technique. Stage II technique is more selective about the text to be analyzed and is better at maintaining original structure of the text.

Sample Output:

<table>
<thead>
<tr>
<th>Base File Stats:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Tags: 7848.0</td>
</tr>
<tr>
<td>Person Names Recorded: 2410.0</td>
</tr>
<tr>
<td>Places Recorded: 2446.0</td>
</tr>
<tr>
<td>Organizations Recorded: 60.0</td>
</tr>
<tr>
<td>Other Names Recorded: 2932.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output File Stats:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Tags: 5558.0</td>
</tr>
<tr>
<td>Person Names Recorded: 2850.0</td>
</tr>
<tr>
<td>Places Recorded: 2541.0</td>
</tr>
<tr>
<td>Organizations Recorded: 167.0</td>
</tr>
</tbody>
</table>

Comparative Analysis (Strict):

| Overall Recall: 0.741253051261188 |
| Person Name Recall: 0.6145228215767635 |
| places Recall: 0.883483237939493 |
| Organizations Recall: 0.033333333333333 |
| Overall Precision: 0.8283555235696294 |
| Person Name Precision: 0.8298245614035088 |
| places Precision: 0.8748524203069658 |
| Organizations Precision: 0.09580838323353294 |
| Overall f-measure: 0.782387493571421 |
| Person Name f-measure: 0.7061267073232196 |
| places f-measure: 0.8791466468726571 |
| Organizations f-measure: 0.049459041731066467 |

Comparative Analysis (Permissive):

| Overall Recall: 0.8835968056863062 |
| Person Name Recall: 0.868750902295294 |
| places Recall: 0.9129076951031162 |
| Organizations Recall: 0.2850000000000003 |
| Overall Precision: 0.8283555235696294 |
| Person Name Precision: 0.8298245614035088 |
| places Precision: 0.8748524203069658 |
| Organizations Precision: 0.09580838323353294 |

Figure 5.2 Stanford NER stats with stage II Text Extraction
Chapter 5 Results

5.2 BERKELEY
The Berkeley Entity system was chosen amongst many reasons, due to its ability to link entities to appropriate Wikipedia entries. With focus on Entity Linking, the experiments were conducted to measure the ease of recognizing and linking entities.

5.2.0 Preprocessing
The first command under the run script for Berkeley is the Preprocessing. It generated a list of tokens with an array of attributes that look like:

<table>
<thead>
<tr>
<th>Filename</th>
<th>Part</th>
<th>Position</th>
<th>Token</th>
<th>POS Tag</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>0</td>
<td>So</td>
<td>RB</td>
<td>(TOP(S* - - - - -)</td>
</tr>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>1</td>
<td>denominated</td>
<td>VBN</td>
<td>(VP* - - - - -)</td>
</tr>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>2</td>
<td>from</td>
<td>IN</td>
<td>(PP* - - - - -)</td>
</tr>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>3</td>
<td>the</td>
<td>DT</td>
<td>(NP* - - - - -)</td>
</tr>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>4</td>
<td>name</td>
<td>NN</td>
<td>* - - - - -</td>
</tr>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>5</td>
<td>Judah</td>
<td>NNP</td>
<td>*)) - - - - -</td>
</tr>
<tr>
<td>part01.txt</td>
<td>0</td>
<td>6</td>
<td>this</td>
<td>DT</td>
<td>(S(NP* - - - - -)</td>
</tr>
</tbody>
</table>

*Table 4.1 Preprocessing output*

The table above represents a section of output file where the columns (in reading order) are Filename, part, position within sentence, token, POS Tag and so-on. The file has not been a part of the study nor has influenced any of the pre-post processing, but indeed plays an important part in running the Berkeley Entity system.
5.2.1 Named Entity Recognition

Named Entity Recognition is a component tool in the Berkeley system that has produced considerably interesting results during the course of study. Once the Berkeley Entity completes the internal preprocessing, it runs the NER system. The NER system makes use of the preprocessed output to generate a similar looking CONLL file that in addition to existing information, consists of NER classes assigned to token identified as Named Entity.

As seen in table 4.2.1.1, the token ‘Judah’ has been correctly tagged as Person.

| part01.txt     | 0 | 0 | So  | RB  | (TOP|S* |   |   |   |   |   |   |   |   |   |
|----------------|---|---|-----|-----|------|---|---|---|---|---|---|---|---|---|
|                |   |   |     |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | denominated | VBN | (VP|* |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | from   | IN  | (PP|* |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | the    | DT  | (NP|* |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | name   | NN  | *    |   |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | Judah  | NNP | *)   |   |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | as     | IN  | (S|BAR|* |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | this   | DT  | (S(NP|* |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | tribe  | NN  | *)   |   |   |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |
| part01.txt     |   |   | obtained | VBD | (VP|VP|* |   |   |   |   |   |   |   |   |
|                |   |   |       |     |      |   |   |   |   |   |   |   |   |   |

Table 4.2 Berkeley NER output
5.2.2 Entity Linking

The Berkeley Entity system, as discussed earlier, consists of many tools including a parser, NER, and wikifier working in a pipeline. The Entity Linking process relies on the pipeline of tools that work before it to create a structure to work upon. The working of the system can be studied from the script of command used to run the system (section 3.3.2).

The results shown in table 4.2.1.1 display the output of last step before Entity Linking starts, i.e. NER and co-reference results. Once this is ready, the Berkeley Entity system uses one of the selected wikifier to generates links between entities available within the wiki dump and Named entities recognized from within the input text.

The output generated by this process is a file with same index for tokens as the two outputs above, sans all other info. Therefore, a 3000 token file, would generate output file with 300 lines and the Entity prediction on each line.

```
(-NIL-* *)*)
*
(Government*
 *
*)
*
*
*
```

In the following excerpt from the output file, the links can be mapped to following tokens:

```
part03.txt 0 7 those DT (NP* - - * (1
part03.txt 0 8 Jews NNPS *)*)*) - - * 26)|1)
part03.txt 0 9 der VBD (VP(VP* - - * -
part03.txt 0 10 the DT (NP* - - * -
part03.txt 0 11 feudal JJ * - - * -
part03.txt 0 12 government NN *)}) - - *
part03.txt 0 13 , , * - - * -
part03.txt 0 14 and CC * - - * -
part03.txt 0 15 was VBD (VP* - - * -
```

Thus showing tokens “the feudal government” to the Wikipedia entry of “government”. Which, considering the underwhelming overall performance of the Entity Linking system, is a palpable match.
5.3 COMPARATIVE

5.3.0 Stanford Vs. Berkeley

While comparing the systems, we look at the standard metrics for Recall and Precision. A concise version of these metrics can be seen in the figures below. For Recall, it can be observed that the mean values are at 51.68% for Berkeley Vs. 45.85% for Stanford. While the median values reflect wider gap with 56.05% for Berkeley Vs. 47.65% for Stanford. The Mean Precision values show yet wider difference with 51.44% for Stanford vs 20.99% for Berkeley. Similarly, the mean f-measures lie at 40.99% for Stanford and 20.77% for Berkeley. Surprisingly the standard deviation is almost the same for recall and precision for both systems, lying in 23%-25% range.
With the two systems (Berkeley & Stanford) yielding interesting set of results, a comparative study was inevitable. After the results from both systems were obtained, the next step in comparative analysis was to transform the outputs in order to assess them and compare the results (as observed in previous sections). The transformation of
the outputs, made the output file into a list of words that could be compared to the base file that would be transformed in similar fashion.

The second step after transforming the three files (base, Stanford NER, Berkeley NER) was to overlay the result files onto base file. This would list to a list of tokens with tags from the base file (manual tagging) plus tags from one of the NER system. The two overlaid files were then used to generate the stats seen in section 4.1 & 4.2.

With the similar stats available for both systems, the final step of the comparative study was to pitch these numbers against each other for a better perspective at their performance on an individual level as well as combined corpus.

Figure 5.6 Line Graph showing recall values for Berkeley & Stanford

Figure 5.7 Line Graph showing precision values for Berkeley & Stanford
5.3.1 NER results for Single-token vs. Multi-token words

After discovering the difference in performance over single token and multi token words. The separate analysis of the two showed following stats:

<table>
<thead>
<tr>
<th></th>
<th>Single Token</th>
<th>Multi Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Recall</td>
<td>68.48%</td>
<td>41.98%</td>
</tr>
<tr>
<td>Mean Precision</td>
<td>89.65%</td>
<td>15.68%</td>
</tr>
<tr>
<td>Median Recall</td>
<td>74.16%</td>
<td>44.38%</td>
</tr>
<tr>
<td>Median Precision</td>
<td>96.14%</td>
<td>4.26%</td>
</tr>
<tr>
<td>Mean F-measure</td>
<td>75.81%</td>
<td>13.22%</td>
</tr>
<tr>
<td>Median F-measure</td>
<td>81.63%</td>
<td>6.44%</td>
</tr>
</tbody>
</table>

*Table 4.3 Statistics for single token vs multi-token NER*

Apart from tag correction, this was another interesting option. It required small changes to scripts that generated the final NER statistics. The scripts were made to calculate the same set of statistics, now for the two categories (Single-token & Multi-token) individually, allowing us to compare the performance of the NER system on the two.

*Figure 5.4 Single token Vs. Multi Token Recall value comparison*
Chapter 5 Results

Figure 5.5 Single token Vs. Multi Token Precision value comparison

To further understand the contribution of each of these categories towards the overall statistics, we look at their composition within each document, i.e. what part of originally tagged tokens are from Single Token words vs. Multi Token words.

Figure 5.6 Histogram for Ratio of tokens from ST vs MT

As seen in the histogram above, of the 276 documents, only 26 documents lie in the range 0.8 to 1.2, i.e. documents with almost equal number of tokens from Single Token words as from Multi Token words. Around 86 have a lot more words from ST vs MT. And the rest (majority) have more tokens from MT words than from ST words, indicating higher influence of Multi Token words on the overall result.
5.3.2 NER performance for each of the 3 classes
In addition to various uses of Stanford results, it was also used to study the performance of NER system individually for each type Entity, i.e. Person, Place, Organization. Following graphs show the Recall and precision values on a scatter plot.

![Recall value Comparison](image1)

*Figure 5.7 Recall value comparison*

![Precision value comparison](image2)

*Figure 5.8 Precision value comparison*
As it can be seen above, it is hard to separate the recall or precision values for person or people names. The performance for organizations though mostly sinks to the bottom of the graph, indicating difficulty recognizing and correctly tagging tokens belonging to words representing organizations.

5.4 Post Processing
With the two black box systems fully functional and used over the complete corpus, the study proceeded with analyzing the output and looking for opportunities to enhance them. One of the most interesting, of these observations was non-uniformity in classifying Named Entity (section 3.5). To verify and further understand this, two experiments were undertaken.

5.4.0 Majority Class Analysis
This analysis looks at the base file to ascertain the uniformity of tags for every token in the file. It involves collection of all tokens that have been tagged at least once. Each token is then accounted for number of times it has been tagged with one or more of the three main classes (person, place, organization). Then we look at the majority class for each token and its proportion. A token with only one class (i.e. its majority class) has a proportion of 1. If a token is tagged with different class for different occurrence in text, the proportion value of majority class is lower 1.

The result of this analysis is a file with token, its total occurrences, its majority class and proportion value. A sample graph below shows the scatterplot for proportion values of tokens from Adams.jews file, sorted by total occurrence.

![Figure 5.9 Scatter plot of proportion values for adams.jews.xml](image-url)
Observations led to the conclusion that the ground truth has consistent tagging and the tokens are aptly tagged with different classes for multiple occurrence cases. Often the reason for ‘a token being tagged with different classes’ was observed to be occurrence of a word both as single token words as well as a part of multi-token word.

5.4.1 Intra-document Tag correction
The intra-document tag correction technique was adopted in response to high proportion of majority class for a major chunk of corpus. With the technique in effect, we observed almost no change in precision with the mean precision values going from 22.36% to 22.24%. But the recall showed a slight improvement with mean recall rising from 51.86% to 52.79%. Although the increase seems really small, further analysis showed a wider range of change with some documents showing up to 5% increase in recall. On Average, the f-measures shot from 20.79% to 21.03%.

Tag correction, as explained in section 3.5, involved tagging all instances of particular token in the NER result with its majority class. This was not done for all tokens but only those with majority class proportion higher than the belief number. The first step of this process is to calculate belief number or qualifying percentage of times a token is tagged with the majority class in NER result. This is very similar to proportion values studied for the base file. Looking at majority class proportion value, it was determined to set belief number at 0.75. This value was later altered to see if the initial threshold held. A 0.75 belief number meant that we believe that tokens that were tagged with a class ¾ times, qualified for extension of majority class to all occurrences. For these tokens, tag correction was implemented, i.e. all occurrences of a token with proportion higher than 0.75 were tagged with its majority class. Tokens with proportion of majority class less than 0.75, were left untouched.

The performance improvement can be seen below in stats.
Figure 5.10 Recall Values Before and After Tag Correction

Figure 5.11 Precision Values Before and After Tag Correction
Chapter 6  ANALYSIS & DEDUCTIONS

6.1  DISCRETE ANALYSIS

6.1.0  Stanford NER
The Stanford NER system gave us the first peek at the NER results on documents from WWO corpus and was a driving factor in the process of modifying the pre-processing technique. The first set of results from Stanford NER with stage I preprocessing, shed light on the Recall/Precision/f-measure for the documents. The following stats were observed:

- Mean Recall – 45.85%
- Median Recall – 47.75%
- Mean Precision – 51.44%
- Median Precision – 51%
- Mean F-measure – 42.99%
- Median F-measure – 44.64%

With the initial figures lying in the range of 60%~80%, the results were considerable. With the initial failure of the Entity Linking, Stanford results provided ground for development and focus on other important area, i.e. NER. Figures from quite a few documents could be considered to be in the league with other widely referred results from Stanford. This proved that historical text, once processed as plain text via correct pre-processing could provide results comparable to text more recently created content.

The next alteration to the results allowed understanding the performance on individual classes. The set of 3 figures was now split into 12 figures, i.e. aforementioned stats for overall, person, place & organization class. This allowed for comparison of NER performance on each class. As per early estimates, the figures were palpable and in aforementioned range for the person and place class. However, organization class had values way lower than the person / place class.

Upon assessment of the input text as well as NER process, it was deduced that organizations were not only difficult to recognize but the lower scores could be credited to other factors like them being multi token words and the difference in training models.
6.1.1 Berkeley NER
The Berkeley NER system results were second set of results available during the study. These set of results were not only indicative of NER performance on historical text, but also allowed for a comparative study (discussed later). The stats looked as following:

- Mean Recall – 51.68%
- Median Recall – 56.05%
- Mean Precision – 22.42%
- Median Precision – 10.71%
- Mean F-measure – 20.77%
- Median F-measure – 14.61%

Looking at the recall and precision, Berkeley NER had figures in the range of 50%~75%. This was lower than Stanford as well as the standard NER expectations. The individual class statistics reaffirm the deductions from Stanford analysis, making organizations a hard match.

Additionally, Berkeley results were also used to analyze single token vs multi token words. This showed that the single-token words had higher precision and recall as compared to the multi-token word (discussed later).

6.1.2 Berkeley Entity Linking
Entity Linking, initially a major part of the study, turned out to be the most excruciating and underwhelming. A large amount of time was spent on setting up and running the Berkeley Entity system. Running an unsupervised system without training had set the expectations low. Surprisingly the system was almost unable to link any entity at the first go. With the adjustments and multiple iterations, the final results showed an output in the range 0~10%. These were the recall figures. Calculating precision would have been a subjective matter where the links would be verified, as there was no ground truth to refer to. Reasons behind the low performance of the system could definitely be credited to the lack of training data and domain specific knowledge. But with the recall being this low, it was impossible to predict or ascertain the extent to which these factors affected the outcome.

Berkeley Entity has some previous results from the CONLL shared task. But due to unavailability CONLL data set, it was hard to reproduce the results in order to verify the status/tuning of the system. This made it further difficult to analyze the output or make any concrete deductions from the output. However, the work done on the WWO corpus in relation to Entity Linking sure paves way for future work and possible experiments, discussed in section 5.4.
6.2 RELATIVE ANALYSIS

This section compares existing results, pitching one technique or system against another. This allows a comparative study to look at the available options and interpret their performance.

6.2.0 Stanford vs Berkeley

As seen in the section 4.3, when the Recall values are compared for the two NER systems, the data is all over the place. It is hard to pick one tool as better than other is. Same can be confirmed by looking at figure 5.1 that shows how difference in recall values for each document are spread over the spectrum and not concentrated on one side of zero.

![Variance in Recall for Berkeley Vs Stanford](image)

*Figure 6.1 Variance in Recall*

When you look at the precision values, Stanford has a clear advantage over Berkeley. Figure 5.2 confirms the same. With only 14 documents having more precise result from Berkeley and about 300+ documents suggesting Stanford a clear winner, the conclusion becomes evident.
6.2.1 Effects of text-extraction techniques (pre-processing)

This analysis looks at Recall and Precision values for all documents, while using two different techniques to extract text. While the analysis suggests that stage II is clearly a better option due to high recall values, the precision values as seen below, seem to be ambiguous.

Figure 6.2 Variance in Precision

![Variance in Precision]

Figure 6.3 Recall for NER with diff Text Extraction techniques

![NER Recall values]
On average, we observed the recall values shoot from 46.91% to 60%. This clearly showed that improving the text extraction can affect the outcome largely and that the structure and format of the input play an important role in NER.

*Figure 6.4 Change in Recall*

*Figure 6.5 Precision for NER using diff Text Extraction techniques*
The precision on the other hand went up from 43.63% to 45.54%. Although the improvement in the precision is not as appreciable as the recall. The combined result, i.e. increase in recall without hampering precision is in itself noteworthy as the two metric tend to affect one another inversely. This can be seen as we look at the F-measures, the means shoot up from 42.91% to 47.49%.

6.2.2 Effects of tag-correction (post-processing)
Here we look at what percentage of dataset is affected by the post processing and how much is the difference in results. As seem below, we can see recall largely showing a positive increase in the range of 0% - 10% of original values. With a detailed look at histogram (figure 5.8) we can conclude that we have over 50% documents showing an increase in recall of 0% - 3% of original values.
Although Figure 5.9 suggests difference in precision in the range of 0-2%. A deeper look at the histogram shows that precision alters from -1% to 1% for majority of documents. If we looked at average change in precision due to tag correction, we would see a null effect.
Figure 6.9 Precision Change due to Post Processing

Figure 6.10 Precision change percentage difference Histogram
6.3 FACTORS OF INFLUENCE

Based on the experiments carried out during the study and various observations made throughout its course, few deductions have been made that generally apply to text processing.

6.3.0 Sentence breaking

While converting XML document to plain text, one feature that made quite a difference is the sentence breaking. Documents in various formats make use of different symbols or structure to indicate end of line, paragraph, sentence, etc. If the document were to be directly ingested by a NER tool, the sentence separation tool would malfunction as they are only programmed to handle plain text where each sentence ends with a ‘.’.

Therefore, it is important identify these various breaks in the text and to put these pieces of text back together, just like they would appear in books. This allows for proper separation of sentences and further proper tagging of words by NER, as both NER systems are known to take sentence structure into consideration for Entity Recognition. This makes sentence breaking, one of the most important factors of influence.

6.3.1 Structure of text (sentences / lists / poems / xml tags)

Another factor that affects the performance of NER system is the structure of text being ingested. As discussed above, we know that sentence breaking and patching is an important process. However, while we look at input documents like a book, apart from the main content or body of the book, other structural components exist, like tables, lists, footnotes, page numbers and so on.

These elements, although a vital part of the book, are of little importance to the NER outcome. The stage I text extraction process, included this during preprocessing, leading to these elements being passed on to pre-processed plain text in some form. However, unfortunately these elements do not fit the traditional structure of plain text, and thus end up not only adding unwanted material that is processed, but also affecting the performance negatively.

It was observed that even after attempting to handle these elements, some of them would end up altering the structure of book-content, often creating line breaks or discontinuity in sentences, thus lowering precision of NER systems.

Stage II text extraction was designed to differentiate the main content of the book from these elements and only process them. As a result, the plain text was visually more readable, continuous and machine ready. The results of NER post change in text extraction proved this with higher precision and recall scores.
6.3.2 Language
The WWO corpus has books from an era that used language that, although English, was a little different from current modern language. Most notably with following differences:

- "i, j, u, v": the language in earlier text had the alphabets ‘i’ in place of ‘j’ and ‘v’ in place of ‘u’. This ends up confusing the POS tagger and ultimately affecting NER results
- Archaic Words: certain words like ‘thee’ are no longer used and are often not a part of training set for NER systems
- Unrecognized Symbols: in addition to words and alphabets, some symbols present in the text are not a part of UTF symbol set. The usual way tools handle this is by replacing the character by its closest relative, like replacing ‘é’ with e. Unfortunately, they aren’t built to handle every possible symbol. Moreover, introduction of an unknown symbol ends up causing the tool to replace it with an ‘unknown’ or random character, further leading to faults in continuity or word recognition.

Overall, these factors affected the text extraction process to transform XML documents into plain text with minimal loss of information. The effect of these factors in evident from the comparative analysis of the two pre-processing techniques discussed in earlier section.

6.4 Further Study

After making the aforementioned observations and deductions, there were points during the study that begged exploration in a direction different from the undergoing experiments or often tasks that required greater amount of time/resources than available. Here we look at some of these topics and explain why they should be the focus of a future study.

- Domain training models / domain knowledge base

The experiments show that the model used to train the NER tools and the corpus documents, do not in particular belong to the same category. Which begs for the question, to what extent can a domain specific model boost the NER results. With a domain specific knowledge base and a training model based on it, it would be possible to experiment and measure this.
• Processing document at a time Vs. multiple-documents together

For the current study, due to a limitation on time and resources, it was only possible to process one document at a time. However, what we have observed through the study is the fact that looking at the document as whole and looking at multiple instances of certain tokens, it is possible to boost the efficiency of the system by replicating successful results. The same phenomenon can be extended to complete corpus if it was possible to process all documents together, allowing cross-document inference.

• Tag-correction (removal of tags for low belief number)

During the process of post processing, it was observed that by extending majority class tag for certain words, it was possible to enhance the performance of the system. Nevertheless, this would only be possible by increasing true positives. Similar technique could be applied in an inverted manner, to remove tags for all instances of a word with high belief number for ‘no tag’. This should in theory allow us to cut the false positives by a few, in turn increasing the precision.

• Cross-document tag correction

Similar to the idea of processing all documents together to allow a corpus wide performance enhancement, it would be interesting to extend the tag correction technique by looking at all instances of a word, not only throughout a document, but also throughout the document. Whether or not this will boost performance beyond intra-document tag correction, can only be determined after such an experiment.

• Better token breaking technique

Unlike the previous few points that looked at improving the post-processing or preprocessing techniques by using them in a different manner, one process indeed requires re-imagining it. Through the course of the study, one intermediate step was found to be malfunctioning at certain stages and could use more work. This was the process of breaking tags for multi-token word, to tag each individual member with the same tag.

In scenarios where the tags wouldn’t end correctly, or there would two classes within same multi-token word, there would be an error where the tag for multi token word would be extended to words after it. This would end up affecting the stats negatively (but not the NER results). Hence, it would be important aspect of a future study to explore a more effective way of breaking tags for multi-token words for a finer assessment of the systems.
Chapter 7  CONCLUSION

This study was commenced to explore the interaction between NLP tools and a specific corpus. The WWO corpus is unlike current newspaper text in many ways, be it its time of origin, its language, or its content. It provided ample challenges, while working with a range of tools like Berkeley, Stanford and more. Yet it can be considered as a representation of ‘Historical text’ in general, allowing for the results of this study to be extended to other datasets that share traits with this corpus. With our set of experiments, deductions and solutions, it would be apt to summarize our findings as following:

- Sentence structure, sentence breaks, unidentified symbols, deviation from standard language model were found to be few facets of preprocessing that affected the NER outcome.
- Enhanced text extraction methods for the XML documents, helped obtain NER results that were palpable, if not superior to existing standards. Moving to a better text extraction technique showed on average 2% better precision and 13% better recall.
- The base text, although served as ground truth, was experimentally found to have differences in NER tags (proportion values lower than 1) for words with multiple occurrences, which meant output metrics would be slightly lower than expected.
- While Berkeley NER showed slight edge over Stanford NER while looking at Recall values, the mean values put this difference at 51% and 45% respectively.
- The difference between the systems was more apparent and inverse while looking at precision. Looking at mean values for precision, Stanford beat Berkeley with 51% against 22%.
- NER results for single token words are in general better than multi-token words. With slight difference in mean recall (68% Vs. 41%) but drastic difference in mean precision (85% Vs. 15%).
- Post processing the NER output with technique such as tag-correction can potentially boost the recall values. Overall, tag-correction raises mean recall by 1% bringing it to 52%.
• An unsupervised, untrained approach at Entity Linking is bound to underwhelm. Study indicates, training a system with domain specific models may show significant increase in performance.

• Entity Linking tools are at a nascent stage of development and not as robust as available NER tools.

The purpose of study was not to excel the existing performance of NER or Entity Linking systems, but to take an untraditional data set as a challenge and explore the difficulties encountered while working with such a data set. This allowed for the establishment of the aforementioned deductions as well as techniques to overcome the barriers that one would encounter while working with a similar set. With this, I hope we have succeeded in helping out future researchers by contributing to the pool of existing knowledge and methods.
APPENDIX A

Sample Script

Sample Script showing instructions for running the Berkeley system.

```bash
# Preprocess the data, no NER
if [ ! -f content2/adams.jews/preprocessed/part1.txt ]; then
    echo "RUNNING PREPROCESSING"
    java -Xmx2g -cp $jarpath edu.berkeley.nlp.entity.preprocess.PreprocessingDriver
        ++config/base.conf -execDir content2/adams.jews/scratch/preprocess -inputDir
        content2/adams.jews/text -outputDir content2/adams.jews/preprocessed
else
    echo "Skipping preprocessing..."
fi

# Now run the joint prediction
if [ ! -f content2/adams.jews/joint/output.conll ]; then
    echo "RUNNING COREF+NER+WIKI"
    # First, need to extract the subset of Wikipedia relevant to these documents. We have already
    # done this to avoid having. Here is the command used:
    #java -Xmx4g -cp $jarpath:lib/bliki-resources edu.berkeley.nlp.entity.wiki.WikipediaInterface -
    #datasetPaths content2/adams.jews/preprocessed -wikipediaDumpPath data/wikipedia/enwiki-
    #lacontent2/adams.jews-pages-articles.xml -outputPath models/wiki-
    #content2/adams.jews.ser.gz
    java -Xmx8g -cp $jarpath edu.berkeley.nlp.entity.Driver ++config/base.conf -execDir
    content2/adams.jews/scratch/joint -mode PREDICT -modelPath models/joint-onto.ser.gz -
    testPath content2/adams.jews/preprocessed -wikipediaPath models/wiki-db-test.ser.gz
    cp content2/adams.jews/scratch/joint/output*.conll content2/adams.jews/joint/
else
    echo "Skipping coref+ner+wiki..."
fi
```
APPENDIX B

Samples from corpus

Here is a sample of text from the corpus for reference.

Below is how the text would have looked. The highlighted part is the rendition of XML code seen above.

Figure A.0.1 Excerpt from adams.jews.xml (document from corpus)

Figure A.0.2 Browser rendition of XML document to replicate the appearance of xml document (text from figure 2.1 highlighted)


6. Shen, Wei, Jianyong Wang, and Jiawei Han. "Entity linking with a knowledge base: Issues, techniques, and solutions." IEEE Transactions on Knowledge and Data Engineering 27.2 (2015): 443-460.


24. https://hridigital.shef.ac.uk/armadillo