1-1-2013

Improving Robustness and Scalability of Available Ner Systems

Amber McKenzie
University of South Carolina - Columbia

Improving robustness and scalability of available NER systems

by

Amber McKenzie

Bachelor of Arts
University of South Carolina, 2003

Master of Arts
University of South Carolina, 2009

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

Manton Matthews, Major Professor

Michael Huhns, Committee Member

Jose Vidal, Committee Member

Anne Bezuidenhout, Committee Member

Thomas Potok, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

Dedication

This dissertation is dedicated to my husband and best friend, without whose love and support I never would have mustered the motivation and perseverance necessary to complete this work. You are my rock.

Acknowledgements

The completion of this dissertation would never have been possible without the continual love and support of my husband, Jay. I will never be able to sufficiently repay you for all you have done for me and only hope that a lifetime of post-dissertation bliss will suffice. Thanks goes to my daughters, Lil and Michaela, who provided me with continual smiles and broke up the monotony of late working nights. I'm sorry you lost a couple of years of your mommy, babies, but I'll now start taking pictures of you again! I am also extremely fortunate to have such a close, supportive family who never wavered in their belief that I would be successful. To Mom, Dad, Sarah, Justin, Allison and the rest of you who believed in me, you will never know how much that love and support meant and strengthened me.

I owe an extreme debt of gratitude to Dr. Hai Ah Nam, who succeeded in pushing me headfirst through the completion of this dissertation, often against my own wishes. She instilled grace and poise where there was whining and tantrums, perseverance when I was ready to quit, and calm strength when frustration was taking over. I have come out a better person for having had the opportunity to be under her tutelage.

I am in constant amazement at my friends who have maintained our friendship up to now because I'm pretty sure I cut myself off from all human contact and came across as crabby and antisocial during most of this process. In particular, I might not have maintained my sanity without the constant support of my friend, Dr. Paul Bogan, who

listened to my rants, commiserated on my failures, talked shop when I needed to brainstorm and offered quiet support throughout. In addition to providing moral and emotional support, Anthony Masi selflessly offered his free time outside of work to provide his services to ease my busywork load and keep me from becoming overwhelmed. Thanks for sticking with me guys. Hopefully I can now get back to being a friend again.

A special thanks to my advisor, Dr. Manton Matthews, the members of my committee, members of the CDA group at Oak Ridge National Lab, and my coworkers at PYA Analytics for ensuring that I put forth my best and for helping me grow into a successful NLP researcher.

Abstract

The focus of this research is to study and develop techniques to adapt existing NER resources to serve the needs of a broad range of organizations without expert NLP manpower. My methods emphasize usability, robustness and scalability of existing NER systems to ensure maximum functionality to a broad range of organizations. Usability is facilitated by ensuring that the methodologies are compatible with any available open-source NER tagger or data set, thus allowing organizations to choose resources that are easy to deploy and maintain and fit their requirements. One way of making use of available tagged data would be to aggregate a number of different tagged sets in an effort to increase the coverage of the NER system. Though, generally, more tagged data can mean a more robust NER model, extra data also introduces a significant amount of noise and complexity into the model as well. Because adding in additional training data to scale up an NER system presents a number of challenges in terms of scalability, this research aims to address these difficulties and provide a means for multiple available training sets to be aggregated while reducing noise, model complexity and training times.

In an effort to maintain usability, increase robustness and improve scalability, I designed an approach to merge document clustering of the training data with open-source or available NER software packages and tagged data that can be easily acquired and implemented. Here, a tagged training set is clustered into smaller data sets, and models are then trained on these smaller clusters. This is designed not only to reduce noise by

creating more focused models, but also to increase scalability and robustness. Document clustering is used extensively in information retrieval, but has never been used in conjunction with NER.

Table of Contents

Dedication ii

Acknowledgements iii

Abstract v

List of Tables ix

List of Figures ix

Chapter 1 Introduction 1

1.1 Problem Description 1

1.2 Research Contribution 6

1.3 Dissertation outline 9

Chapter 2 Background and related work 11

2.1 Existing NER systems 12

2.2 Available systems and data 28

2.3 Feature aggregation 31

2.4 Domain adaptation 35

2.5 Clustering for NER 38

2.6 Document clustering 42

2.7 Active learning 44

Chapter 3 Document clustering 47

3.1 Top similar documents 50

3.2 K-means 54
3.3 Topic modeling 55
3.4 TF-ICF and cosine similarity 56
3.5 Cluster adaptation 58
Chapter 4 Improving robustness and versatility 60
4.1 Tagger versatility 60
4.2 Available data sets 64
4.3 Annotation optimization 68
Chapter 5 Improving scalability 74
5.1 CoNLL 74
5.2 CoNLL and Ontonotes 76
Chapter 6 NER recommendations 79
6.1 Data 80
6.2 Clustering 81
6.3 Tagger 82
6.4 Performance and scaling 83
Chapter 7 Conclusion 84
7.1 Future work 86
References 89

List of Tables

Table 2.1 – Performance impact of the inclusion of a variety of different baseline features from the work of Zhang and Johnson (2003). Table definition below results 19

Table 2.3 – NER F1 on the dev set and test set, using different representation trained on RCV1. Some word representations were induced over the cleaned RCV1, as indicated by the second column. C&W is (Collobert & Weston, 2008). 25

Table 2.4 – F1 scores for Ratinov and Roth (2009) NER system comparing BIO and BILOU labeling formats tested on both CoNLL03 and MUC7 datasets. 28

Table 2.5 – Feature aggregation results tested on CoNLL03, MUC7 and web pages data sets from Ratinov and Roth (2009) 33

Table 2.6 – Example results of bootstrapping technique from Sun and Grishman (2011), including Brown bit string representation used to traverse binary tree to produce hierarchical clusters (Section 2.5) 37

Table 3.1 – K-means cluster results; details F1 score of the test system, the performance of the model training on the entire training set, and the make-up of the clusters. 54

Table 3.2 – Topic model cluster results; details F1 score of the test system, the performance of the model training on the entire training set, and the make-up of the clusters. 56

Table 3.3 – TF-ICF model cluster results; details the original test system F1 score, the F1 score after augmenting the clusters (Test+), the performance of the model training on the entire training set and the cluster make-ups after augmentation. 57

Table 4.1 – Comparison of F1 scores between LBJ and Stanford taggers trained on full training sets 62

Table 4.2 – F1 scores of Stanford tagger model trained on full training set compared with cluster-based models 63

Table 4.3 – F1 scores for Ontonotes clusters using LBJ tagger compared with model trained on entire Ontonotes data set 66

Table 4.4 – F1 scores for top three Ontonotes clusters with combined smaller clusters using LBJ tagger compared with model trained on (a) entire Ontonotes data set (7351 documents) and (b) smaller top Ontonotes data set 67

Table 5.1 – Performance of clustering technique compared to training a model using the full training set for both the original training documents and doubled training sets. 75

Table 5.2 – Clustering technique compared to training a model using the full CoNLL training set for both the original training documents and quadrupled training sets using Stanford tagger. 76

Table 5.3 – Clustering technique compared to training a model using the full CoNLL and Ontonotes training sets using Illinois tagger. 77

Table 5.4 – Clustering technique compared to training a model using the full CoNLL and Ontonotes training sets using Stanford tagger. 78

List of Figures

Figure 1.1 – Example NER tagger output 2

Figure 1.2 – Diagram of current NER system framework 3

Figure 1.3 – Example of ambiguous NER tagger output 4

Figure 2.1 – Syntactic (TB), lexical (KC) and layered representations from Goldberg et al. (2009) 21

Figure 2.2 – Sample syntactic word clusters, each column displaying the top 10 words in one cluster and their probabilities from Li and McCallum (2005) 22

Figure 2.3 – Sample semantic word clusters, each column displaying the top 10 words in one cluster and their probabilities from Li and McCallum (2005) 22

Figure 2.4 – Example (a) BIO vs. (b) BILOU tagging 27

Figure 2.5 – Semantic clusters created using Brown clustering taken from (Brown, deSouza, Mercer, Della Pietra, & Lai, 1992) 39

Figure 2.6 – Sample clustering of words for one class in the Wall Street Journal corpus taken from (Ushioda, 1996) 40

Figure 2.7 – Examples of clusters of cliques and their associated contexts taken from (Ah-Pine & Jacquet, 2009) 42

Figure 3.1 – Diagram of approach 49

Figure 3.2 – Number of documents for which each system achieved better F1 scores. ... 51

Figure 3.3 – Average percentage points better and worse in the F1 score that the proposed system achieved compared to the standard LBJ tagger for models trained with the top 20, 50, 100, and 300 similar documents. 52

Figure 4.1 – Screenshot of tool to facilitate NER text annotation 69

Figure 4.2 – F1 score trends using document ordering compared to unordered training sets 73

Chapter 1 Introduction

1.1 PROBLEM DESCRIPTION

In the last decade, the world has become immersed in digital information, necessitating a complete transformation in the way that we handle this information. 1.8 zettabytes, or 1.9 billion terabytes, of digital information was created by the world in 2011, with a projected 7.7 zettabytes in 2015 according to the IDC market projection (Gantz & Reinsel, 2011). Any industry or organization that relies on information has had to reevaluate their internal processes and update technologies to be able to analyze and incorporate the new data medium and the overwhelming quantity that comes with it. A notable example of organizations that have acutely felt the impact and challenges of this shift to digital data is law enforcement. Whereas law enforcement officers used to simply collect physical evidence when building a case, much of the evidence is now in digital format on suspects' computers, phones, external hard drives and cloud storage. In addition to a change in evidence medium comes a sharp increase in the amount of data that is accumulated and must be analyzed during the course of a forensic investigation. Cases can involve many sources of data, totaling many terabytes in a single case for forensic analysts to examine. Analyzing this amount of data by hand is unfeasible and can lead to mistakes or missed critical information. To further compound the problem, the data can be extremely varied, ranging from technical manuals or academic papers to emails or chat records.

Named entity recognition (NER) is a robust field within natural language processing (NLP) that has come to play a large role in text-based digital data analysis in a number of applications, such as question answering or document retrieval systems. NER is a subset of NLP processes called sequence-labeling tasks, meaning that for a given sequence of tokens, each token will be tagged with a certain label depending on the task. For the NER task, once a data set has been tagged, this information can be used for further NLP tasks including relation detection and text summarization. Specifically, NER aims to identify, extract, and classify proper names within text data. This facet of NLP has a number of applications, the most prevalent of which is for information extraction (IE). The goal of IE is to automatically extract pertinent pieces of information from a given text-based dataset. This information may be used for information retrieval, question answering systems or to populate a knowledge base. Often, a large portion of the information that is needed from text consists of entities such as people, places, and organizations that contribute a significant amount of meaning to the information contained in the data set. For example, users searching for relevant news articles will query for entities such as Barack Obama, New York, or Microsoft, shown in figure 1.1. It is these types of entities that are targeted by NER systems.

President [PER Obama] spoke with the CEO of [ORG Exxon Mobil Corporation], [PER Rex Tillerson], about oil drilling in the [LOC Caspian Sea].

Figure 1.1 – Example NER tagger output

The basic framework of current statistical NER systems, depicted in figure 1.2, involves training a machine learning model to predict named entities within a given text. A set of sample data, annotated with the correct tag (classification) for each word or token, is used as the training data for the NER system, which generally utilizes a machine learning algorithm to generate a predictive model based on a variety of features of the data. Once this model has been created, it can be used to tag new data and output the probable named entities that are contained within those documents.

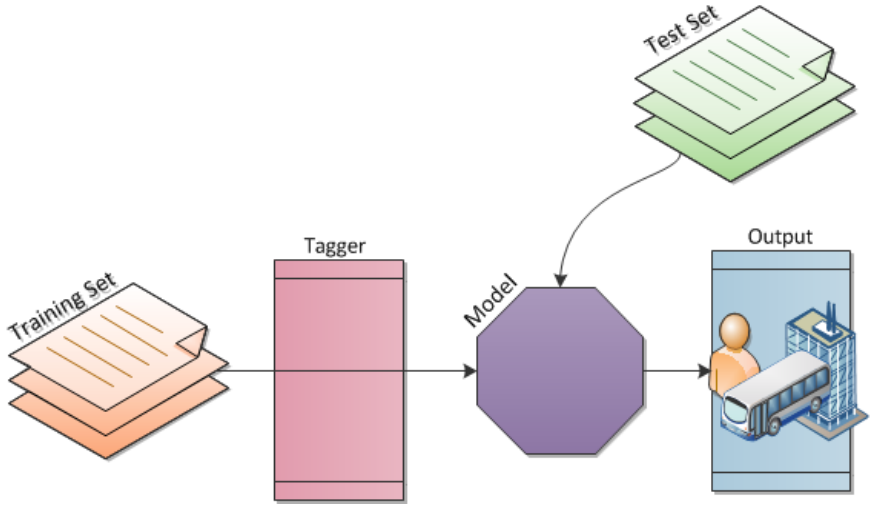

Figure 1.2 – Diagram of current NER system framework

In law enforcement applications, NER systems can make a substantial impact to ameliorate the data deluge, as investigators are most interested in finding information that pertains to specific entities and desire to extract these entities computationally rather than by hand due to the significant amount of diverse digital evidence that must be examined. Though research in the area of NER is fairly extensive, current state-of-the-art solutions are generic, succeeding only for domains similar to their training data, and still fail to

adequately provide functionality that is adaptable to a broad range of domains (Tkachenko & Simanovsky, 2012). Natural language can contain many instances of non-standard grammar and spelling and ambiguous wording that can make some sentences difficult, even for humans, to understand. This can cause significant difficulty in conducting this analysis computationally, such as in the example in figure 1.3, where the use of the word "Washington" needs to be properly differentiated for the three tags.

Mr. [PER Washington] took a late flight out of [LOC Washington, D.C.] on his way to [LOC Seattle] to meet with the owners of [ORG Washington Appliances, Inc.]

Figure 1.3 – Example of ambiguous NER tagger output

Another domain in which out-of-domain text poses significant problems for sequence-labeling tasks is that of technical manuals or maintenance records. Although NLP researchers have conducted a plethora of NER domain adaptation research in an attempt to develop systems that will achieve better accuracy on out-of-domain data, the resulting systems are either custom made and not open source, or they require additional data sources in the form of tagged target data or external data. In previous research, part-of-speech (POS) tagging and information extraction, other sequence-labeling tasks within NLP, were investigated for helicopter maintenance records with the intent to extract usable information to further the efforts of condition based maintenance (McKenzie, Matthews, Goodman, & Bayoumi, 2010). While many IE systems have already been developed for use in analyzing a variety of different types of texts, the majority of these systems are developed to analyze documents written in Standard English, such as news

articles and literature. These systems are not suitable for use with maintenance data, given the informal and often ambiguous nature of the language used in these reports. For this application, a custom system involving a hierarchy of multiple POS taggers and a number of hand-crafted text-chunking rules had to be created to produce the desired results and even required further adaptations when faced with changing data. This research demonstrated the limitations involved with standardized training sets and domain-specific NLP systems.

In order for an open-source NER systems to be broadly effective in a data analysis application, such as in a forensic investigation, the system must be robust enough to handle large amounts of varied data. For data that is significantly different from that used to train a standard, generic model, it may be necessary to manually tag domain-specific data with which to train a more focused model. In this instance, tagging a large amount of data – such as that used to train a generic model – is not feasible due to time and cost constraints for the average forensic investigator, and much less possible for every new case. The standard CONLL 2003 training set, used by many current NER systems, contains over 900 tagged documents. Assuming an expert can tag a document in 30 minutes, the expert would require over 450 hours to reproduce a training set of that size from the target domain data. Even assuming an optimistic 15 minutes per document would result in 225 man hours to create the tagged corpus. This scenario also assumes the availability of a qualified person with knowledge of NER and how the data should be tagged. Many organizations and law enforcement agencies do not regularly employ NER or NLP researchers who would be able to tag domain-specific training data or develop a tailor-made NER system to achieve better performance than that of an open-source

system trained on a generic training set. Further, these organizations are often constrained by the amount of computing resources that are available to them and do not have access to high-performance computing platforms that would enable easier scalability and faster training times.

In the face of the challenges impacting time- and money-constrained (TMC) organizations lacking NLP resources, new techniques must be developed to adapt existing NER tools and tagged corpora for better performance without specialized assistance. The goals and motivation of the research are shaped by the needs and requirements of TMC organizations amidst an increased need for expeditiousness and a desire to do computational, instead of manual, analysis in the face of an overwhelming amount of data that is continually growing. For this research, we aim to develop a new approach that adapts existing NER systems and tagged data sets for more efficient use without having to design a specialized tagger, manually tag additional data, or utilize high performance computing hardware to complete the computational requirements.

1.2 RESEARCH CONTRIBUTION

The focus of this research is to study and develop techniques to adapt existing NER resources to serve the needs of broad organizations without expert NLP manpower. My methods emphasize usability, robustness and scalability of existing NER systems to ensure maximum functionality to a broad range of organizations. Usability is facilitated by ensuring that the methodologies are compatible with any available open-source NER tagger or data set, thus allowing organizations to choose resources that are easy to deploy and maintain and fit their requirements. In law enforcement, an agency might have to rely on existing NER systems, given the alternative of spending excess money and significant

time to have an expert tag data manually for every new case. This major hurdle has motivated the use of available training sets in the place of domain-specific tagged data.

Pre-built systems trained on a standard corpus of news articles contain numerous named entities that either would not likely appear in digital forensic evidence or occur in contexts that do not provide useful feature information, resulting in poor accuracy. Without specialized systems or domain-specific tagged data, the only options for performance improvement for TMC organizations or law enforcement agencies is to increase the robustness of existing systems and optimize the use of the available training sets to produce models that are more effective at tagging domain-specific data. For out-of-domain target data, more source training data does not necessarily imply better accuracy (McKenzie A. , 2013). Because the training data is not of the same domain as the data to be tagged, it contains a lot of noise in the form of tagged entities that do not apply to the target data. My approach focuses on increasing the robustness of available systems and resources while reducing noise, maintaining usability and improving scalability.

One way of making use of available tagged data would be to aggregate a number of different tagged sets in an effort to increase the coverage of the NER system. However, this poses challenges in terms of scaling, as model generation complexity increases exponentially as training sets get larger. The larger the training set containing these noisy and irrelevant named-entity examples, the less focused the resulting model will be and the less accurately it will tag the digital forensic data. Augmenting the training data with more generic tagged data contributes to the creation of a more complex model that is less pertinent to the given task domain. In particular, larger training sets can

dilute the efficacy of the context aggregation feature component of many tagger designs. These systems pull information about the contexts in which named entities occur within the training data and use this data to enhance the NER model. When aggregating over a given window of data, the more generic the training data, the more generic the obtained feature information will be because relevant entity examples will be more spread out across the data and are less likely to fall within that window. Though, generally, more tagged data can mean a more robust NER model, extra data also introduces a significant amount of noise and complexity into the model as well. Because adding in additional training data to scale up an NER system presents a number of challenges in terms of scalability, this research aims to address these difficulties and provide a means for multiple available training sets to be aggregated while reducing noise, model complexity and training times.

In an effort to maintain usability, increase robustness and improve scalability, I designed an approach to merge document clustering of the training data with open-source or available NER software packages and tagged data that can be easily acquired and implemented. Here, a tagged training set is clustered into smaller data sets, and models are then trained on these smaller clusters. This is designed not only to reduce noise by creating more focused models, but also to increase scalability and robustness. Document clustering is used extensively in information retrieval, but has never been used in conjunction with NER.

To continue with the previous example, a law enforcement organization wants to conduct NER on each case that comes in. Rather than waste precious man hours manually tagging a new training set, existing tagged data – either found online or

organizational data that had been previously tagged – is clustered into a number of smaller, more focused groups. These groups then become the training sets used to generate the same number of NER models. Each document to be tagged is clustered into the group it is most similar to and tagged using the model trained with that group's training documents. In this way, a document is tagged with a model that is likely to have more useful and relevant features. For each new case, forensic analysts must only measure the similarity between the incoming data's documents and the training clusters and do not need to tag more data. To further increase the robustness of such a system, analysts can employ the developed annotation tool to greatly facilitate the creation of domain-specific tagged data. In general, the introduction of document clustering is designed to improve the robustness and scalability of existing NER systems.

1.3 DISSERTATION OUTLINE

In support of the ideas presented in the previous sections, Chapter 2 presents a background section that includes general information about general NER and integral concepts, as well as the most relevant and recent research innovations in the area. Related work includes work in the areas of statistical NER, features and word representations, feature aggregation, available data and NER taggers, domain adaptation, clustering techniques, clustering in NER and active learning. Chapter 3 details document clustering experiments conducted to test their viability for inclusion in an NER system. Chapter 4 establishes the flexibility of the techniques involved in the approach by highlighting its portability to different taggers and data sets and the integration of the developed annotation tool. Chapter 5 examines the performance advantages achieved by the approach. The recommendations for use of this approach in a real world setting are set

forth in Chapter 6. Conclusions and possible avenues for future work are presented in Chapter 7.

Chapter 2 Background and related work

Named-entity recognition (NER) is a subtask of information extraction (IE), whereby structured information is automatically extracted from unstructured or semi-structured machine readable documents. Using NER methods, one seeks to process human language texts using natural language processing (NLP) to locate and classify elements of text in predefined categories such as the names of persons, organizations, and locations. Some of the challenges in NER include the extensive annotation labor and ensuring robust performance across domains. This research aims to improve existing statistical NER systems to address both of these challenges by integrating document clustering to develop better, more focused models that can be employed across many domains. Clustering the training set will directly impact the effectiveness of the feature aggregation component of a statistical NER system and will alter the resulting model that is used to make the tag predictions. As Dalton et al. note, "Another area that could be improved is a more principled approach to selecting the passage collection to use for feature expansion" (Dalton, Allan, & Smith, 2011). Research on domain adaptation details work striving to increase the robustness of NER systems. These techniques either adapt what data is used to train the model or the underlying system itself in an attempt to improve performance on out-of-domain data. Document clustering for NER, on the other hand, has the potential to improve robustness of existing systems using available tagged data.

A brief background on the important components comprising a standard NER system is given in Section 2.1, including recent work on developing and improving NER systems. Following an explanation as to why statistical techniques were chosen over rule-based systems, a general discussion of statistical systems is provided, including details on model generation, feature selection and word representations. Section 2.2 and beyond describe areas of NER research related to the use of document clustering techniques, including feature aggregation, domain adaptation, clustering and active learning, and highlight the shortcomings of current techniques and methodologies in those areas.

2.1 EXISTING NER SYSTEMS

This section provides a brief discussion on NER and why statistical NER and why statistical NER systems are generally considered state-of-the-art for the field. In Section 2.1.2, an overview is given of features, or characteristics, of text-based data that are typically used to predict the classification (tag) of a given word. Features combined with the word representations, (presented in Section 2.1.3), or the way that text and its features are represented to be able to encode more data in a more compact manner, impact statistical model development and the efficacy of that model in identifying named entities within text. Finally, Section 2.1.4 provides information about the output format of an NER system with some examples.

2.1.1 Statistical NER

Research on NER approaches falls into two categories: rule-based systems and statistical techniques. Rule-based NER involves finding patterns within the data's morphology or syntax that provide clues as to a word's category. Gazetteer-based

systems fall into this category, as they provide look-up tables of token-entity pairs. Because syntax, morphology and word choice vary greatly in today's digital data – e.g. academic publications vs. tweets vs. technical manuals, etc. – rules that are based on features of a given data set cannot necessarily be transferred to a new domain. While some of the more generic features (discussed in section 2.1.2) are incorporated into statistical models, rule-based systems themselves are not well-suited for domain adaptation given their reliance on the training data itself. Changing target domains would mean that many rules would likely no longer apply and the system would have to be adapted with new domain-specific rules, or a new gazetteer manually compiled, for every new data set. Given this limitation, the majority of current NER research surrounds statistical techniques. While acknowledging that a large body of research exists on rule-based NER, the research for this thesis focuses on statistical NER due to its adaptability and potential for broad ranging applicability.

Research on statistical NER can be divided into three general categories – supervised, semi-supervised and unsupervised – which describe the amount of human interaction involved in the training of the NER system.[1] Ideally, an unsupervised system is desired but, in general, providing more human input in the form of annotated data from the target domain or fine-tuning system parameters often proves to out-perform unsupervised techniques. In statistical NER methods, a machine learning (ML) algorithm is trained using a data set that in which each word has been given an appropriate tag and produces a model that can then be used to make inferences over future data. Learning methods take feature vectors as training input in order to learn information about the data

[1] For a more comprehensive discussion of statistical NER, see (Jurafsky & Martin, 2009).

that will provide guidelines on how to infer tags of testing data. These feature vectors are composed of a set of parameters, generally unique to the target data set, whose values are set based on the given token, word or document that they represent. In this way, text data can be expressed by way of vectors that can be easily processed by existing ML techniques. Likewise, feature vectors can be used to encode knowledge from large amounts of unlabeled text without the unwieldy necessity of processing the text itself through the algorithm. Once a data set has been analyzed for features and represented by a feature vector, these vectors are aggregated and probabilities for sequence labels are computed based on occurrences within the document set. The resulting model is used to provide a prediction as to the label of a specific token based on the previous occurrences of the token and its surrounding tokens in the training data or its features and characteristics. With respect to model generation, a number of factors must be taken into account, including learning model choice and inference model algorithm.

Supervised and semi-supervised approaches require a previously-tagged data set that is used to train a model which is then used to predict tags for previously-unseen data. The language models used in these approaches are a function for determining the conditional probabilities used in predicting a given output - e.g. tags, words, or documents - based on the prior input. These models are closely tied to the task that they are assigned to, with one language model performing well on one task, such as machine translation, but not on another, such as semantic role labeling. These models do not necessarily have to stand alone or be mutually exclusive. Many researchers integrate several different types of models to increase coverage of their system and improve performance. For their work, Uszkoreit and Brants combine a partially class-based model

- using the results of their distributed word clustering implementation - with a word-based model for use in a state-of-the-art system for machine translation leading to improvements in translation quality (Uszkoreit & Brants, 2008). Griffiths et al. investigates how to model both short- and long-range dependencies within a document by implementing a mixture of models: one that models syntactic relations among word classes and one that models semantic correlations between words in and across documents and found it to be competitive in part-of-speech tagging and classification tasks (Griffiths, Steyvers, Blei, & Tenenbaum, 2005). Many times, language models can be easily combined, and if it becomes difficult to represent the full variety of desired features with one generated model, the process might be broken down to better capture the nuances of named entity features.

In addition to model choice, learning algorithm choice is also a consideration in supervised NER system development, as there are a number of different ML algorithms available and each produces models that perform differently. In their overview of recent work on NER, Nadeau and Sekine note that hidden Markov models (HMMs), decision trees, maximum entropy (maxEnt) models, support vector machines (SVMs) and conditional random fields (CRFs) have all been used as supervised learning algorithms for NER (Nadeau & Sekine, 2009). Ratinov and Roth, reporting the best performance to-date on the CoNLL-2003 shared task dataset, employed a regularized averaged perceptron, another type of machine learning algorithm, for their NER system (Ratinov & Roth, 2009). The two NER systems used in the experiments for this research incorporated a CRF and a perceptron into their framework (Finkel, Grenager, & Manning, 2005) (Ratinov & Roth, 2009).

Supervised and semi-supervised techniques and approaches, while popular, are not conducive to achieving the development of a successful unsupervised NER system, in which no human input is required. However, many of the recent unsupervised approaches are extremely limited in scope, focusing on a single targeted domain, or use techniques that are not conducive for use in specialized domains that may not contain entities arising in Wikipedia. The unsupervised system developed by Usami et al. was equipped only to handle biomedical data, in which the corpus is only tagged with one semantic class, Gene or Gene Product (GGP) (Usami, Cho, Okazaki, & Tsujii, 2011). Munro and Manning developed an unsupervised system that relies on a set of unaligned parallel texts in different languages (Munro & Manning, 2012). Lin et al. extract entities based on Wikipedia Infoboxes in different languages (Lin, Snover, & Ji, 2011). With the explosion of web-based data freely available for use, in particular data that includes categorical information such as in Wikipedia, a number of researchers have chosen to use this data to fuel their NER engine (Urbansky, Thom, Schuster, & Schill, 2011), (Szarvas, Farkas, & Ormándi, 2007), (Janik & Kochut, 2008). Domain-specific and external data-dependent unsupervised systems are not practical when trying to extract entities from a large variety of, possibly esoteric, data. Domain-specific systems, such as those for biomedical data, would perform poorly if applied to data from a different domain, such as general emails. In order for it to be successful with other data sets, adaptations would have to be made to the system, whereby eliminating the desired unsupervised aspect. While a strictly unsupervised system is not likely to be successful in an area requiring domain adaptation, clustering training documents in a semi-supervised approach could allow for existing, out-of-domain training sets to be utilized to better success without human intervention.

Document clustering can be combined with statistical techniques in an effort to introduce additional unsupervised elements into the NER process.

2.1.2 Features

Statistical NER models rely on characteristics, or features, of the words and their surrounding contexts to provide the information needed to be able to make future predictions about the classifications of words. Features must be informative about the data they are representing so that learning methods can make models that can adequately predict tags. The most common features concern lexical information likely because that type of information is the most obviously identified and extracted. The one-hot representation includes the most basic feature: the word itself. However, these types of representations that include only basic information about the word itself cause problems with data sparsity because the nature of language is such that many words are hardly, if ever, seen in training data. Allison et al. investigate the data sparsity problem in relation to large amounts of data and confirm that large numbers of words from a vocabulary will not be represented in even significantly large data sets (Allison, Guthrie, & Guthrie, 2006). In order to allow the model to make predictions for previously unseen words and also to reduce the sparseness of the model, more complex linguistic features – such as information regarding the word's morphology or syntax – and features regarding the contextual instances of the word or word type within the data set should be included. These types of features also facilitate the construction of a more abstract model that is less domain-specific.

2.1.2.1 Baseline

Zhang and Johnson focus their NER system development on determining which types of features work best for NER and in which combinations (Zhang & Johnson, 2003). This work establishes a group of features that have been proven to be useful for the NER task to serve as a baseline set for theirs and future systems. They divided their feature set up into two categories. Simple token-based features included the token itself, prefix and suffix information, and capitalization. More sophisticated linguistic features included part of speech (POS) and chunking tags and four dictionaries[2]. They found that the actual token itself does not have a significant impact on NER performance. Though the word is not particularly useful, prefix and suffix information, as well as capitalization, saw significant impact on NER performance. Table 2.1 highlights their findings in terms of the performance impact of various feature combinations, with the reference feature description described below the results table. POS and chunking features produced little improvement, and the dictionaries supplied a "small, but statistically significant improvement" (2). The inclusion of several additional dictionaries derived from external sources was also tested, though they were not part of the baseline features. These dictionaries proved to further boost performance of the system.

[2] The four dictionaries were the ones supplied for the CoNLL-2003 shared task.

Table 2.1 – Performance impact of the inclusion of a variety of different baseline features from the work of Zhang and Johnson (2003). Table definition below results

Experiment ID	Features used	Precision	Recall	$F_{\beta=1}$
1	A+C	91.94	74.25	82.15
2	B+C	93.70	74.89	83.25
3	A+F	89.96	82.50	86.07
4	B+C+D	88.79	86.01	87.38
5	B+C+D+E+F	90.11	88.67	89.39
6	B+C+D+E+F+G+H	91.00	89.53	90.26
7	B+C+D+E+F+G+H+I	92.14	90.73	91.43
8	B+C+D+E+F+G+H+I+J	92.76	91.42	92.08

Feature ID	Feature description
A	Tokens that are turned into all upper-case, in a window of ± 2.
B	Tokens themselves, in a window of ± 2.
C	The previous two predicted tags, and the conjunction of the previous tag and the current token.
D	Initial capitalization of tokens in a window of ± 2.
E	More elaborated word type information: initial capitalization, all capitalization, all digitals, or digitals containing punctuations.
F	Token prefix (length three and four), and token suffix (length from one to four).
G	POS tagged information provided in shared the task.
H	chunking information provided in the shared task: we use a bag-of-word representation of the chunk at the current token.
I	The four dictionaries provided in the shared task: PER, ORG, LOC, and MISC.
J	A number of additional dictionaries from different sources: some trigger words for ORG, PER, LOC; lists of location, person, and organizations.

Representing some of the most extensive recent work on NER, Ratinov and Roth cite their baseline features, based on the work by Zhang and Johnson, as being the previous two predictions in the sequence, the current word, the current word type, the prefixes and suffixes of the current word, the five-word window that includes two words before and two words after the current word, the pattern of capitalization in the five-word window sequence and the bigram of the current word and the previous tag (Ratinov & Roth, 2009).

These two sets of baseline features are fairly representative of common practices in NER systems. Most systems go further in trying to add in extra features that will significantly boost their performance. However, it is important to determine what additional features provide the ideal tradeoff between acquisition costs and performance benefits for the system.

2.1.2.2 Feature types

Though the actual features do not differ significantly, different research approaches classify features in a variety of different ways. For example, features can be categorized by their location or generation within the data set or by their linguistic classification. Chieu and Ng give a detailed description of the features that they use for their maximum entropy NER approach (Chieu & Ng, 2003). They break their extensive feature list into three categories: local features from the sentence containing the word, global features about the other occurrences of the word in the document and features derived from gazetteers.

Goldberg et al. discuss their integration of syntactic and lexicon-based features (Goldberg, Tsarfaty, Adler, & Elhadad, 2009). In developing their parser, they found that different resources - a tagset and a lexicon/morphological analyzer - contained different sorts of linguistic information and did not want to try to reduce one to the other. Instead, they propose to produce a fuzzy mapping between the two resources - the "morphosyntactic-transfer layer" – which they surmise captures the interaction between the two representations. Their layered approach is illustrated in figure 2.1. Though they talk about their feature sets in different terms than Chieu and Ng, they are still referring

to standard features such as POS tags, prefix/suffix information, and lexicon (vocabulary).

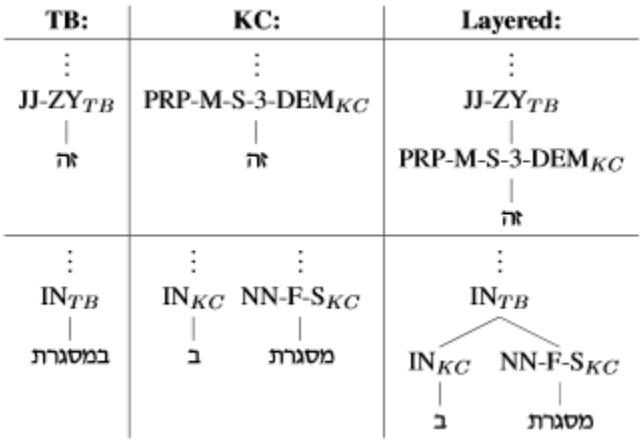

Figure 2.1 – Syntactic (TB), lexical (KC) and layered representations from Goldberg et al. (2009)

Li and McCallum refer to their groupings as function and content words and developed a model that could distinguish between the two types, thereby identifying syntactic and semantic categories (Li & McCallum, 2005). Figure 2.2 provides examples of syntactic word clusters, while figure 2.3 shows the words occurring in most frequently in semantic clusters. They aim to have different cluster features for a word in different instances of that word in the document.

make	0.0279	of	0.7448	way	0.0172	last	0.0767	to	0.6371
sell	0.0210	in	0.0828	agreement	0.0140	first	0.0740	will	0.1061
buy	0.0174	for	0.0355	price	0.0136	next	0.0479	would	0.0665
take	0.0164	from	0.0239	time	0.0121	york	0.0433	could	0.0298
get	0.0157	and	0.0238	bid	0.0103	third	0.0424	can	0.0298
do	0.0155	to	0.0185	effort	0.0100	past	0.0368	and	0.0258
pay	0.0152	;	0.0096	position	0.0098	this	0.0361	may	0.0256
go	0.0113	with	0.0073	meeting	0.0098	dow	0.0295	should	0.0129
give	0.0104	that	0.0055	offer	0.0093	federal	0.0288	might	0.0103
provide	0.0086	or	0.0039	day	0.0092	fiscal	0.0262	must	0.0083

Figure 2.2 – Sample syntactic word clusters, each column displaying the top 10 words in one cluster and their probabilities from Li and McCallum (2005)

bank	0.0918	computer	0.0610	jaguar	0.0824	ad	0.0314	court	0.0413
loans	0.0327	computers	0.0301	ford	0.0641	advertising	0.0298	judge	0.0306
banks	0.0291	ibm	0.0280	gm	0.0353	agency	0.0268	law	0.0210
loan	0.0289	data	0.0200	shares	0.0249	brand	0.0181	lawyers	0.0210
thrift	0.0264	machines	0.0191	auto	0.0172	ads	0.0177	case	0.0195
assets	0.0235	technology	0.0182	express	0.0144	saatchi	0.0162	attorney	0.0161
savings	0.0220	software	0.0176	maker	0.0136	brands	0.0142	suit	0.0143
federal	0.0179	digital	0.0173	car	0.0134	account	0.0120	state	0.0138
regulators	0.0146	systems	0.0169	share	0.0128	industry	0.0106	federal	0.0138
debt	0.0142	business	0.0151	saab	0.0116	clients	0.0105	trial	0.0126

Figure 2.3 – Sample semantic word clusters, each column displaying the top 10 words in one cluster and their probabilities from Li and McCallum (2005)

2.1.3 Word representations

Word representations, which can encode lexical and linguistic information about the word and its surrounding context and usage, can be used as a means to compute similarities between words and can therefore be used to generate a model that will be able to make predictions for words not used in the construction of that model. The use of word representations for NLP tagging tasks allows for more flexibility and possibilities in system design and performance. Likewise, techniques for extracting word representations automatically from text have provided the means to expand the set of possible features for NLP tagging tasks by enabling more information to be included without significant human effort. Previously, semantic information about words or tokens most often had to

be hand encoded because that information was not available at the word level and had to be provided externally from the system. This led to the development of domain-specific systems that must be tailored to fit their target data set. Facilitating the generation of feature vectors encoded with semantic representations diminishes the necessity of domain adaptation and might lead to more robust systems that can process a wider variety of data types.

While including word representations in NLP systems is a step in the right direction toward advancing the field, simply including random features about the text will not provide a significant benefit. Tishby et al. introduce the information bottleneck method for finding the optimal tradeoff between accuracy and complexity in extracting information about a given dependency (Tishby, Pereira, & Bailek, 1999). Their ideas apply to signal, as well as text, processing, and they assert that it is important to understand what information plays a role in predicting some output in order to specify the best function to do the prediction. This idea has significant implications for NLP and suggests that different types of features are likely more beneficial for certain NLP tasks than others. When developing a system for a specific NLP task, a set of features must be identified that optimizes performance for that given task.

Word representations benefit an NER system in a variety of ways. They allow for more information to be encoded into a model, and varying the information included in the word representations will produce differences in the performance of the ML algorithm when applied to the input data. Word representations can allow us to utilize powerful supervised ML algorithms that can have more predictive power than many non-statistical techniques, depending on how well the model is generated. Word representations can be

automatically generated from unlabeled data, which introduces unsupervised learning into the system and reduces the amount of necessary human interaction. This automatic generation facilitates the discovery of more abstract features, not readily discernible by humans, which help develop domain-independent NER systems.

Though many features can be hand generated, this can be time consuming and a confining method for the task in that humans are limited in the observations we can make about a data set. Being able to induce word features automatically from unlabeled data introduces a number of possibilities and flexibility to an otherwise labor-intensive task. So-called unsupervised word representations have become popular in recent NER research and have the ability to facilitate work towards the development of an effective domain-independent, unsupervised NER system.

Unsupervised word representations can generally be categorized into three different groups: distributional, distributed and clustering. Distributional representations involve an aggregation of the information concerning the co-occurrence of words across a given context (Turian, Ratinov, & Bengio, 2010). In contrast, distributed representations, also referred to as word embeddings, involve multiple dimensions that represent latent features of a word. Clustering-based representations involve clustering words together and using inclusion in a cluster as a class label. Turian et al. conducted an investigation of these word representations for the NLP tasks of NER and chunking (Turian, Ratinov, Bengio, & Roth, 2009). Distributional representations are not considered in the experiments in this overview because the authors claim that there is a lack of research on this type of representation for sequence labeling tasks resulting in uncertainty as to what settings would be best for applying distributional representations to these tasks. Through

numerous experiments of different combinations of word representations and NER systems, they concluded that cluster-based word representations performed the best on the NER task, as seen in Table 2.2 taken from (Turian, Ratinov, Bengio, & Roth, 2009). However, the work did not consider distributional representations, did not include any soft clustering representations, and did not focus on out-of-domain performance.

Table 2.2 – NER F1 on the dev set and test set, using different representation trained on RCV1[3]. Some word representations were induced over the cleaned[4] RCV1, as indicated by the second column. C&W is (Collobert & Weston, 2008).

Representation	Clean	Dev F1	Test F1
Brown clusters	✓	92.47	88.39
Brown clusters		91.65	88.10
HLBL embeddings		92.32	87.66
C&W embeddings	✓	91.77	87.13
C&W embeddings		91.14	86.27
none (baseline)	n/a	89.87	84.07

The use of automatically generated word representation as included features for NER has implications for the development of both unsupervised and domain-independent NER systems. They allow for a more robust system to be developed because the generated features tend to represent a more general and abstract nature of the words, allowing the features to be applied to a broader domain space. These word representations also allow for unsupervised aspects to be combined with machine

[3] RCV1 is taken from the Reuters corpus and is a superset of the CoNLL '03 data set.

[4] Here, "cleaned" refers to eliminating any sentences of which less than 90% of the letters are lowercase.

learning algorithms to produced unsupervised or semi-supervised techniques, where supervised prevailed before.

In order to best utilize these approaches, the most efficient way for generating the word representations must be identified so that the benefits gained in domain-adaptation and reduction in human interaction from their use are not counter-balanced by loss in accuracy and performance. Also, it should be investigated as to whether different generation techniques are more appropriate for domain-independent NER and whether these techniques can be customized to be more generalizable.

2.1.4 NER output

Given that NER is a sequence-labeling task, the choice of labeling format has come under consideration by some NER researchers. Once a data set has been tagged, each token within the data set is assigned a tag that denotes whether or not it is an entity and if so, what type. The most basic tagging convention is to simply tag entity tokens with their entity type – e.g. person, organization, location – and tag all other words with the standard "O" designation. However, these labels do not serve to indicate whether sequences of like tags are the same entity and provide a minimal amount of information in the form of features when training a model. A step further, and the most popular type of labeling, is BIO, where the 'B' stands for the beginning of an entity, 'I' is in or inside an entity and 'O' is outside or not part of an entity. The 'B', 'I', or 'O' is then followed by the entity type. In this way, entities can be extracted in chunks rather than by single tokens. This tagging convention is employed by the majority of current NER systems and researchers. In an effort to improve NER models by expanding the amount of information

provided by NER tags, Ratinov and Roth investigated the usefulness of another tag set, BILOU (Beginning, Inside, Last, Outside, Unit-length) (Ratinov & Roth, 2009). Figure 2.4 (a) and (b) demonstrate the differences between the two labeling conventions, though both would extract the same entity chunks.

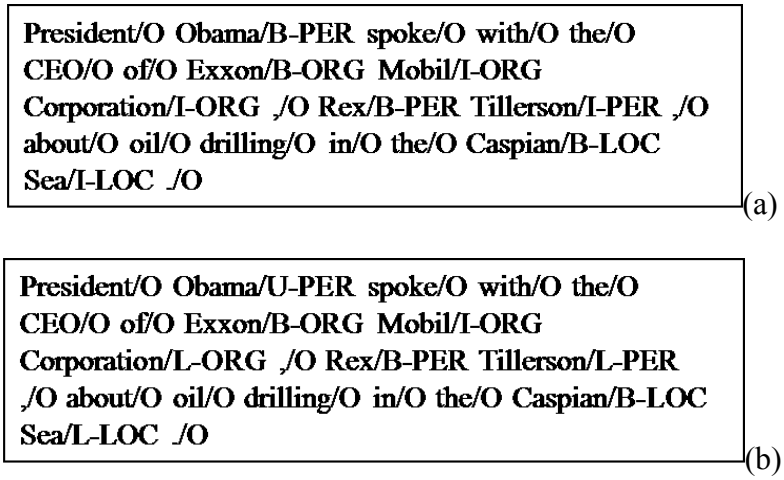

Figure 2.4 – Example (a) BIO vs. (b) BILOU tagging

Ratinov and Roth determined that BILOU outperformed BIO for the NER tagging task. The results of these experiments are depicted in Table 2.3. This tagging set is used in a couple of recent NER systems, but most still use BIO due to the fact that it has become the standard for many available systems (Ratinov & Roth, 2009) (Usami, Cho, Okazaki, & Tsujii, 2011) (Finkel, Grenager, & Manning, 2005) (Ritter, Clark, Mausam, & Etzioni, 2011).

Table 2.3 – F1 scores for Ratinov and Roth (2009) NER system comparing BIO and BILOU labeling formats tested on both CoNLL03 and MUC7 datasets.

Rep.	CoNLL03		MUC7	
Scheme	Test	Dev	Dev	Test
BIO	89.15	**93.61**	86.76	85.15
BILOU	**90.57**	93.28	**88.09**	**85.62**

The set of categories assigned to entities also varies by system and is dictated by the training set. Categories must be semantically relevant and, while there are a handful of generally generic tags such as person, organization and location, many tags are more domain specific. For example, the system of Usami et al. for tagging biomedical data incorporates only one semantic class, Gene or Gene Product (GGP) (Usami, Cho, Okazaki, & Tsujii, 2011). Models incorporate the tag set as features provided by the training data and/or gazetteers. The Stanford tagger offers three model options, with a three- (location, person organization), four- (location, person, organization, misc), or seven-category tag set (time, location, organization, person, money, percent, date) which are based on different training sets (Finkel, Grenager, & Manning, 2005). Ratinov and Roth have also experimented with different tag sets with varied success (Ratinov & Roth, 2009).

2.2 AVAILABLE SYSTEMS AND DATA

To make the benefits of NER research broadly applicable to organizations without NLP specialists, this work focuses on improving existing NER data sets and taggers, which can be found online. For this research, I investigate several data sets and two NER

taggers, chosen based on their availability, ease of use and rate of previous use by other researchers.

2.2.1 Data

A number of NER tagged data sets have been created and made available online. Standard NER research uses the Conference of Natural Language Learning (CoNLL) 2003 data set as a baseline benchmark. This corpus was developed and distributed as part of the CoNLL-2003 shared task for language-independent named entity recognition systems. Each year CoNLL has a challenge task to stimulate research and development of NLL systems. The 2003 shared task was the last of the conference devoted to NER. Researchers generally test their developed NER systems on this data set, and it is regarded as the standard for determining the success of their NER techniques. The English portion of the data comprises newswire data from the Reuters Corpus and can be obtained free from NIST. The training set contains 946 documents, while the test set contains 231. The use of this data set with NER research is necessary to compare the technique with previous NER approaches.

The Ontonotes 4 data set is the result of a project between BBN Technologies, the University of Colorado, the University of Pennsylvania and the University of Southern California's Information Sciences Institute, the goal of which was to tag a large corpus composed of 7351 documents from a number of different genres: news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, and talk shows (Hovy, Marcus, Palmer, Ramshaw, & Weischedel, 2006). This data set is available through the Linguistic Data Consortium for the price of shipping. Tkachenko and Simanovsky use

this data, along with CoNLL 2003, as one of their benchmark data sets in their exploration of features for NER (Tkachenko & Simanovsky, 2012).

Most other available sets are from the early 2000s and late 1990s, though there are a few domain-specific sets, such as for twitter or biomedical data that have been tagged more recently. A variety of other data sets are available, some free and some not. A twitter data set was tagged and made available by Ritter et al. for their work on NER in tweets (Ritter, Clark, Mausam, & Etzioni, 2011). They tagged 800 randomly sampled tweets using a tag set based on the open-domain ontology Freebase. The MUC 7 data set from the Message Understanding Conference in 1997 is another that has been used by some NER researchers and is a subset of the North American News Text Corpora (Ratinov & Roth, 2009). The MUC 3 and 4 data sets are also available but are from even earlier in the 1990s, which will not effectively represent recent language change. The availability of these data sets makes it feasible to do NER research without resorting to expending a great deal of time manually tagging a new one.

2.2.2 Taggers

One of the advantages to the approach proposed in this research is that a specialized tagger is not required to implement the proposed techniques. Furthermore, no modifications must be made to existing systems. I present studies using two of the most commonly used and best open source NER taggers are those produced by the Cognitive Computation Group at the University of Illinois at Urbana-Champaign and by the Stanford Natural Language Processing Group. Both are considered to be state-of-the-art generic taggers in the field. The Illinois Named Entity Tagger (LBJ), part of their

Learning Based Java software suite, relies on gazetteers, word class models from unlabeled text, and non-local features to produce their models (Ratinov & Roth, 2009). The Stanford tagger takes its cues from machine learning in its use of a conditional random fields (CRF) classifier augmented by Gibbs sampling "a simple Monte Carlo method used to perform approximate inference in factored probabilistic models" (Finkel, Grenager, & Manning, 2005). These taggers are relatively simple to download, install and get running.

Some other software packages include NER taggers in their options. The General Architecture for Text Engineering (GATE) out of the University of Sheffield includes a built-in information extraction component, ANNIE, which contains a semantic tagger component (Cunningham, Wilks, & Gaizauskas, 1996). The Natural Language Toolkit (NLTK) also contains a built-in MaxEnt named entity tagger (Bird, Klein, & Loper, 2009). There are also several other NLP packages that provide NER capabilities; however, in general, these packages take longer to set up, have a more significant learning curve, and do not allow for as much adaptation and modification as the Stanford and LBJ taggers. They are also suited for more basic NER and are not designed for tougher NER problems, such as that encountered in digital forensic investigation. For these reasons, the LBJ and Stanford NER taggers were a more suitable choice for use with this research.

2.3 FEATURE AGGREGATION

Restructuring the training documents in the manner detailed for this research will have a direct impact any aspect of the system that depends on the ordering of these training documents, such as feature aggregation. Feature aggregation refers to collecting

feature information from across a document or document set, rather than simply taking the information from a particular word instance. The method of acquiring features has become an integral part of building an NER prediction model. Because aggregating the context of every named entity across an entire training set can be fairly computationally expensive and introduces significant noise into the features due to the many contexts in which an entity may occur, many researchers have chosen instead to conduct local aggregation, such as across a document, or with a certain window of tokens that may span several documents. However, this method leaves the choice of context to chance: determined by how the documents are organized within the training set. A better option would be to choose the context that best represents the entities to be tagged. Current research attempts to refine the methods of feature aggregation or manipulate the contexts, but none focuses on altering the training sets as a means for improving feature aggregation.

Ratinov and Roth, whose research details their work on the University of Illinois NER tagger, provide a number of different feature aggregation approaches in their discussion of design considerations for NER (Ratinov & Roth, 2009). They refer to the information gathered from aggregation as non-local features and categorize the different approaches as context aggregation, two-stage prediction aggregation and extended prediction history. Context aggregation refers simply to aggregating the context that tokens appear in across a given document. Two-stage prediction involves applying a baseline NER system to the training documents and use the resulting labels as features for those given tokens. In an effort not to treat all tokens in a text similarly, which they assert is the case with context aggregation and two-stage prediction, Ratinov and Roth

developed an approach for non-local feature generation based on extended prediction history (Ratinov & Roth, 2009). Their approach is based on the idea that named entities are easier to spot at the beginning of texts where they are first introduced. Table 2.4 details the results of their experiments where they keep track of all label assignments for the token in the last 1000 words and use that probability information as a prediction history feature for the token. On testing the performance of their NER system with the three feature aggregation approaches, the authors concluded that the approaches are complementary and that no single approach out-performed the others.

Table 2.4 – Feature aggregation results tested on CoNLL03, MUC7 and web pages data sets from Ratinov and Roth (2009)

	Component	CoNLL03 Test data	CoNLL03 Dev data	MUC7 Dev	MUC7 Test	Web pages
1)	Baseline	83.65	89.25	74.72	71.28	71.41
2)	(1) + Context Aggregation	85.40	89.99	**79.16**	71.53	70.76
3)	(1) + Extended Prediction History	**85.57**	**90.97**	78.56	**74.27**	72.19
4)	(1)+ Two-stage Prediction Aggregation	85.01	89.97	75.48	72.16	**72.72**
5)	All Non-local Features (1-4)	86.53	90.69	81.41	73.61	71.21

Krishnan and Manning introduce a two-stage approach to feature aggregation (Krishnan & Manning, 2006). They implement a layered approach of two classifiers based on CRFs in which the second uses the output of the first as features. In addition to a set of standard baseline features, the occurrences of tokens, entities and entities that contain other entities (so named "superentities") are aggregated over both documents and the entire corpus, resulting in a set of six additional features, in an effort to construct a soft-constraint label consistency. By applying a soft constraint using document and corpus aggregation, the authors strive to encourage identical labeling of same entities, but

not make it a requirement, thus remaining flexible for the possibility of different types of labeling for the same entity in the case of entities that might be ambiguous.

Huang and Yates present their feature aggregation approaches in the form of smoothing of the dataset (Huang & Yates, 2009). Their goal for smoothing is the same as for aggregation in that they strive to extend the usefulness of the model by sharing information about multiple contexts for a token in order to provide more information about words that are rarely, or never, seen in training. In experimentation, the authors found that their smoothing approach improved performance on rare words, out-of-domain text, and smaller training sets.

Dalton et al. take an external knowledge approach to context aggregation (Dalton, Allan, & Smith, 2011). Using an information retrieval method called Pseudo-Relevance Feedback (PRF), they query for relevant passages in an external data set using the context for the target token. Given that they searched for the context that the entity occurs in, it is assumed that the top returned passages all contain instances of the entity with the same label. They then aggregate the features for this token across a number of the top retrieved documents and induce features based on this information. Their approach is compared with the Stanford NER and LBJ NER systems and found that their aggregated features improved performance over those systems.

With feature aggregation, researchers strive to expand the context used to predict the classification of a given token. Much of the recent work on features for NER has been related to aggregation of some sort in an effort to widen model coverage, decrease human interaction in the feature generation process, and increase detection and classification accuracy. Many systems incorporating feature aggregation have seen performance

improvements over other nearly state-of-the-art systems. However, only Huang and Yates and Dalton et al. make an effort to make changes to the input data used to train the model. With feature aggregation being so dependent on the supplied context, more research must be devoted to determining what optimizations can be made with regards to the training data so as to improve the feature aggregation portion of the system.

2.4 DOMAIN ADAPTATION

Because semi-supervised systems, in which the efficacy of the predictive models is determined by the inputted training data, are so prevalent in current NER research, the majority of systems are largely dependent upon the domain through the training set used to generate the NER model. These currently-available systems exhibit poor performance on out-of-domain data in general. Liu et al. applied the Stanford NER system, considered to be one of the best NER systems currently available, to a data set of tweets and found that the performance dropped from the 90.8% achieved on the CoNLL03 shared task data set to a dismal 45.8% average F1 score on the out-of-domain data (Liu, Zhang, Wei, & Zhou, 2011). Dalton et al. tested the same system on a corpus of historical books and only achieved 51% accuracy (Dalton, Allan, & Smith, 2011). Other researchers have tested NER systems trained with data from one domain on data from another and demonstrated deteriorated performance using common NER algorithms (Rüd, Ciaramita, Müller, & Schütze, 2011). Futher, Ciaramita and Altun trained a HMM model on the Reuters corpus using a perceptron algorithm and test it on the out-of-domain Wall Street Journal (WSJ) test set (Ciaramita & Altun, 2005). They observed a drop in F-measure from 91% on the Reuters test set to 64% on the WSJ test set. This previous research

demonstrates that domains that do not already have tagged data available can make only limited use of mainstream NER systems for their applications.

Because of this poor performance, many researchers strive to develop techniques or systems that will achieve better performance on out-of-domain data. Domain adaptation approaches in NER have involved including additional data or employing methodologies to adapt NER models to better address target data (Ben-David, et al., 2010), (Rüd, Ciaramita, Müller, & Schütze, 2011), (Guo, et al., 2009), (Wu, Lee, Ye, & Leong, 2009), (Sun & Grishman, 2011). However, all of these approaches involve hand-tagging or using external data, or creating specially-designed systems that are not freely available for use by other researchers or organizations.

Ben-David et al. use a small amount of tagged data from the target domain combined with a larger amount of available out-of-domain tagged data to improve tagger performance (Ben-David, et al., 2010). However, this approach does not prove suitable for applications in which the target data changes frequently. In the work of Rüd et al., search results similar to the target entity are used to extract additional features with which to augment and adapt the NER model to the target data (Rüd, Ciaramita, Müller, & Schütze, 2011). The motivation for their work was to apply a system trained on news articles to web query data. In this instance, the approach is only truly applicable to the web query domain, as the additional features are extracted from this domain and will likely not transfer well to other domain-specific data.

Guo et al. employ latent semantic association to fine tune a NER model without including any additional domain-specific training data (Guo, et al., 2009). Their system learns latent semantic association among words from untagged text, which is then used to

augment or tune the original model, with the idea that words in different domains will still share similar semantic and syntactic contexts. Bootstrapping, or using the output of a system to refine and improve the system itself, is a common approach to the domain-adaptation problem in NER. Wu et al. combine traditional bootstrapping ideas with domain adaptation goals to select instances that are both informative and easy to automatically label correctly (Wu, Lee, Ye, & Leong, 2009). They also set criteria that stop the bootstrapping process before it begins to add in incorrectly labeled instances. In this way, they aim to identify and incorporate instances that contain both domain-independent and target-domain specific features. Similarly, Sun and Grishman employ bootstrapping in their system but also include additional features based on membership in Brown word clusters generated from both source and target data (Sun & Grishman, 2011). Example clusters from their bootstrapping process are presented in Table 2.5, demonstrating how words are grouped together by the clustering process and thus provided with a classification to be included as an added feature.

Table 2.5 – Example results of bootstrapping technique from Sun and Grishman (2011), including Brown bit string representation used to traverse binary tree to produce hierarchical clusters (Section 2.5)

Bit string	Examples
110100011	*John, James, Mike, Steven, Dan, ...*
11010011101	***Abdul, Mustafa, Abi, Abdel**, ...*
11010011111	***Shaikh, Shaykh, Sheikh, Sheik**, ...*
1101000011	*President, Pope, Vice, ...*
111111110	***Qaeda, Qaida, qaeda, QAEDA**, ...*
00011110000	*FDA, NYPD, FBI, ...*
000111100100	***Taliban**, ...*

2.5 CLUSTERING FOR NER

Clustering of training documents has not been previously used in NER research; however, the general idea of clustering has been implemented to improve NER systems and models. Word clustering is the most common application of clustering in the NER space though some recent work has incorporated clustering of just the named entities, instead of all words in a corpus. Word clusters are a common addition to semi-supervised NER models in current research. Unlabeled data is clustered and membership in those clusters is included as an additional feature for supervised learning. In this way, even if words are not in the training data, if they share characteristics with a cluster, they will still be able to be classified.

Much of the early word-clustering work approaches the problem in one of two ways: either words are moved around among groups until some ending condition is met or clusters are repeatedly merged until a satisfactory partitioning is reached, generally one in which the average mutual information (AMI)[5] is maximized. However, most recent research in clustering tends to follow the merging approach, which was pioneered by Brown et al. (Brown, deSouza, Mercer, Della Pietra, & Lai, 1992). The work by Brown et al. was motivated by the need to make predictions on a string of text from a noisy channel and the desire to assign words to classes based on a large body of text. In this approach, each word in the vocabulary of the training data starts out in its own cluster. Clusters are repeatedly merged based on which merging will produce the least amount of loss of AMI. In this way, they strive to find the clustering that maximizes the

[5] Mutual information is a measure of how much information one variable can provide about another, or the mutual dependency of two variables.

amount of information the model contains about the target domain. Once a partitioning is achieved to reach the desired number of classes, reshuffling of words sometimes can improve the AMI of the model. Figure 2.5 demonstrates some of the clusters obtained from a sample text from the Canadian parliament using the Brown clustering technique. This idea of clustering similar words into class designations is used extensively in subsequent research on NLP labeling tasks with many researchers seeking to make improvements which will increase performance and accuracy of their systems.

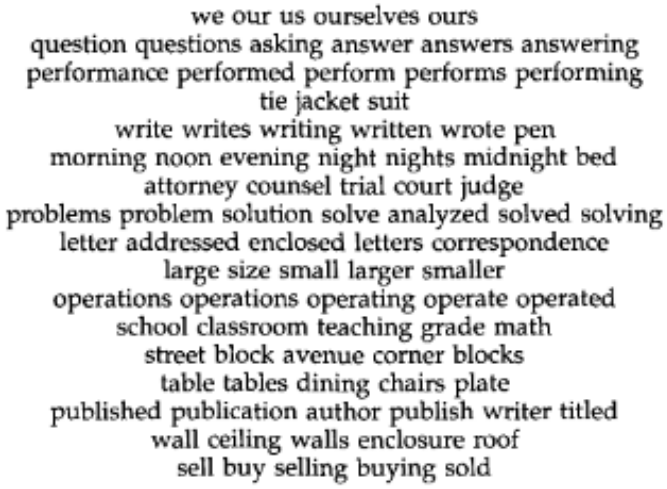

Figure 2.5 – Semantic clusters created using Brown clustering taken from (Brown, deSouza, Mercer, Della Pietra, & Lai, 1992)

Usioda introduced a hierarchical clustering of words in an effort to improve their decision-tree based POS tagger in their parsing system (Ushioda, 1996). He also attempts to combine the two ways of clustering - shuffling between clusters and merging clusters - to determine if that can improve performance. The author asserts that clusters provide more functionality for the system if they can be constructed at variable granularities or in

a hierarchically structured way. Further, clusters should promote what he calls "mutual substitutability" in which clusters can represent both syntactic and semantic information. A sample word clustering is given in figure 2.6. This extends the capabilities of the system to more competently handle unknown words. Lin and Wu take word clustering one step further by also including phrase-based clustering (Lin & Wu, 2009). They also implement the algorithm in such a way so as to enable scaling up to tens of millions of clustering elements.

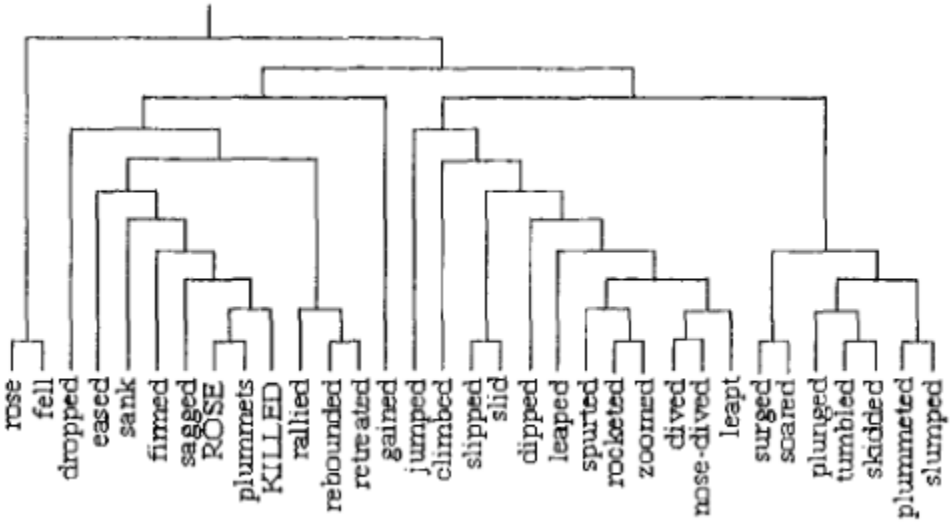

Figure 2.6 – Sample clustering of words for one class in the Wall Street Journal corpus taken from (Ushioda, 1996)

Whereas most previously mentioned clustering implementations use a hard clustering methodology, or one in which every word can belong to only one cluster, Li and McCallum employ a soft clustering technique in which words can probabilistically belong to multiple classes (Li & McCallum, 2005). Lin and Wu describe an extension to their k-means algorithm that can convert their hard-clustering implementation to a soft-

clustering (Lin & Wu, 2009). Instead of just adding a word to the most appropriate cluster, the extension would allow a word to be added to any cluster within a given similarity value. This produces a sort of fuzzy mapping between words and clusters and could prove to be more flexible when developing an NER system for wider coverage.

Koo et al. test a two stage approach in which they establish clusters based on unlabeled data and then pass those clusters to a discriminative learning algorithm to identify informative features (Koo, Carreras, & Collins, 2008). In this way, they learn features without requiring prior information as to the origin of those features. Koo et al. were able to show that clustering can reduce the need for supervised data by half. Though their work was targeted for dependency parsing, the ideas can be applied to other NLP tasks, such as sequence labeling tasks, as evidenced by the work of applying discriminative learning to NER conducted by Miller et al. (Miller, Guinness, & Zamanian, 2004).

In contrast to word clustering, Ah-Pine and Jacquet cluster cliques of named entities in order to identify other possible annotations for a given entity (Ah-Pine & Jacquet, 2009). Some example clusters of cliques and the contexts they are associated with are provided in figure 2.7. Their motivation is to resolve ambiguities and incorrect annotations output by a given NER system. The use of clustering in NER is limited in its application and has the potential to provide a higher degree of usefulness if utilized in other areas such as training document clustering.

num clu	more significant NEs		more significant contexts	
4	Oxford_NOUN	497	1.be_VERB.AT	77.17
	London_NOUN	291	1.area_NOUN.MOD	63.56
	Liverpool_NOUN	252	1.have_VERB.AT	50.66
	Manchester_NOUN	240	1.move_VERB.TO	48.23
	Newcastle_NOUN	166	1.member_NOUN.FOR	44.76
	Leeds_NOUN	135	1.magistrate_NOUN.MOD	42.19
	Edinburgh_NOUN	131	1.go_VERB.TO	41.91
	Birmingham_NOUN	125	1.live_VERB.IN	41.47
	Glasgow_NOUN	123	1.be_VERB.NEAR	41.05
58	Cambridge_NOUN	26	1.study_VERB.AT	8.76
	Oxford_NOUN	26	1.professor_NOUN.AT	8.25
	London_NOUN	7	1.student_NOUN.AT	7.27
	Edinburgh University_NOUN	6	1.graduate_NOUN.MOD	7.24
	Edinburgh_NOUN	5	1.attend_VERB.AT	6.06
	Oxford University_NOUN	5	1.be_VERB.AT	5.93
	Westminster_NOUN	4	1.degree_NOUN.MOD	5.70
	Glastonbury_NOUN	4	1.teach_VERB.AT	5.62
	Cheltenham_NOUN	4	1.educate_VERB.AT	4.88
95	Wembley_NOUN	11	1.beat_VERB.AT	4.71
	Ibrox_NOUN	10	1.play_VERB.AT	4.51
	Twickenham_NOUN	9	1.final_NOUN.AT	4.27
	Elland_NOUN road_NOUN	6	1.win_VERB.AT	4.13
	Highbury_NOUN	5	1.match_NOUN.AT	4.00
	Oxford_NOUN	5	1.game_NOUN.AT	3.52
	Wimbledon_NOUN	4	1.face_VERB.AT	3.49
	Cheltenham_NOUN	4	1.crowd_NOUN.AT	3.18
	Ascot_NOUN	3	1.the_DET game_NOUN.AT	2.84

Figure 2.7 – Examples of clusters of cliques and their associated contexts taken from (Ah-Pine & Jacquet, 2009)

2.6 DOCUMENT CLUSTERING

Document clustering has been used extensively in machine learning, with many approaches developed for clustering documents. The clustering methods perform with varying degrees of success for different applications. This observation, combined with the lack of previous research applying document clustering to NER, means that it was necessary to test a number of different clustering approaches to determine the optimal clustering strategy for use in this setting. As with domain adaptation, significant research has been devoted to this area, and many algorithms and systems are developed for a specific purpose or application area. For this research, k-means, topic modeling and a

clustering technique based on cosine similarity are investigated as options for inclusion with NER systems due to their widespread general use and the availability of existing code and tools.[6]

K-means is a common clustering methodology employed in many machine learning applications. Simply, *n* documents are assigned to *k* clusters based on the similarity of their vector representation to the mean of that cluster. It is an iterative procedure in which after documents are assigned to clusters, cluster centers are determined and these new cluster centers are then used to reclassify the documents. Given that k-means represents one of the more commonly used approaches, it was a logical initial choice for testing (Steinbach, Karypis, & Kumar, 2000). In order to conduct a clustering of the documents, they must first be converted to a representational format from which similarity can be measured, most often a vectorized form. Term frequency – inverse document frequency (TF-IDF) is a common document representation protocol in which the frequency of each term is related back to its frequency across the documents in a corpus, giving an indication of the importance of the word within the corpus (Robertson, 2004).

Another method of conducting document clustering called topic models are utilized as a means of representing the semantic content of a document, rather than simply using the standard bag-of-words representation, in which a vector is created out of all of the words in the lexicon and documents are represented based on which words they contain (Steyvers & Griffiths, 2007). Topics consist of clusters of words that generally

[6] An in-depth discussion of document clustering is out of the scope of this work. For a more detailed explanation, refer to Shah and Mahajan's work (Shah & Mahajan, 2012).

occur together and are a means of highlighting abstract concepts contained in a document. Once a corpus has been statistically analyzed for potential topics and documents have been assigned to a number of topics based on their similarity to those topics, clusters can be generated by grouping documents from the same topic or topics.

As a third methodology, documents were represented by term frequency – inverse corpus frequency (TF-ICF), an alternative to TF-IDF that utilizes observations based on Zipf's law to provide a corpus based estimate of TF-IDF (Reed, et al., 2006). TF-ICF is a good choice for out-of-domain NER because the base corpus is generic and not dependent on the given data set. Document vectors were compared using cosine similarity and clustered into groups based on a specified similarity threshold (Reed, Potok, & Patton, 2004). Previous work has shown that cosine similarity was effective when choosing top similar documents and could likely be effective for this application. Using this implementation, the user is able to alter the threshold of similarity between documents in the clusters. In this way, the technique performs a form of hierarchical clustering, another common document clustering approach.

2.7 ACTIVE LEARNING

In the event that human resources are available to create a domain-specific tagged training set, active learning has become increasingly popular as a means of decreasing the amount of tagged data required to create an efficient NER model. Active learning refers to the idea of using machine learning algorithms to choose the data to learn from, ultimately resulting in the need for less data to be used (Settles, 2009). Olsson presents an extensive survey of active learning as it relates to natural language processing (Olsson, A

literature survey of active machine learning in the context of natural language processing, 2009).

With reference to the specific task of named entity recognition, Shen et al. attempt to maximize the usefulness of the information provided to the model by a given example based on three criteria: informativeness, representativeness and diversity (Shen, Zhang, Su, Zhou, & Tan, 2004). They employ a support vector machine to choose examples based on the quantified measures they developed for the three specified criteria. These techniques were able to reduce labeling costs by 80% without showing significant reductions in performance. Becker and Osborne pursue a committee-based approach in which a number of different classifiers are implemented, each taking into account a different feature space (Becker & Osborne, 2005). The degree of deviation of the classifiers determines whether an instance is potentially interesting and deserves further examination by a human annotator. In the work of Vlachos, active learning is compared to the coined term "active annotation", in which data is tagged using an unsupervised tagger, the resulting data used to train a model and that model is used to identify the instances to be fed to the human annotator (Vlachos, 2006). Kim et al. explore an adaptation to uncertainty-based systems in the form of an entropy-based measure for quantifying the classifier's uncertainty (Kim, Song, Kim, Cha, & Lee, 2006). They also strive for diversity within their sampling set and combine these two goals using the MMR (Maximal Marginal Relevance) method to rank the potential samples. Olsson introduces a bootstrapping approach to named entity annotation (Olsson, 2008). in which a set of documents is manually annotated, this set is used as a seed for machine learning to

identify future documents to tag, and the remaining documents are pre-tagged using the resulting system, while using a human annotator to conduct corrections.

While all of these active learning methods were able to significantly reduce the time and effort required by human NER annotators, some visible limitations still remain. Given the reliance of the approaches on machine learning, the methods require a significant investment in terms of implementation, as the majority of machine learning algorithms can be rather complex. Many of the best active learning algorithms are closely coupled to the machine learning algorithm being utilized. Future instances or documents being fed to the algorithm are chosen based on their degree of informativeness for the model. However, what constitutes this informativeness is contingent on the particular algorithm, thus introducing a level of dependence and specialization to the implementation.

Chapter 3 Document clustering

Manually tagging large training sets or developing customized systems are not viable solutions for mainstream NER needs, such as for companies and organizations who do not have the time and money to develop their own system or acquire additional data, as is often the case in law enforcement. The focus of this research is to improve the robustness, scalability and time-to-solution of existing NER resources without resorting to developing custom, application-driven systems. Document clustering techniques present a promising option for creating smaller, focused training sets that allow for larger overall training data sets and greater scalability and have previously never been used in this manner. To explore the effects of document clustering, I investigated several different clustering techniques using the CoNLL 2003 data set to determine which is best suited for the NER application area. The three clustering techniques that were explored – k-means, topic modeling, and cosine similarity – were chosen due to their diversity within the field and the availability of a simple implementation.

The CoNLL 2003 data used comprised the test (not the development) and training documents in the CoNLL-2003 shared task data. This corpus is considered to be the baseline standard for most current NER research and is a necessary inclusion in order to make the experiments comparable to other research in the field. Also, it has been noted that the test and training sets within the corpus are not as similar in nature as are the development and training sets (Ratinov & Roth, 2009). The training set contains 946

documents, while the test set contains 231. The NER tagger produced by the University of Illinois at Urbana-Champaign, one of the best performing systems on the CoNLL 2003 data set, uses a 1000 token window across which to take their global context aggregation (Ratinov & Roth, 2009). For this system, the F1 score using one model trained on all 946 training documents was 90.77. By choosing 1000 tokens, Ratinov and Roth hope to be able to capture a large enough example set to provide a robust feature value while maintaining a reasonable computation time. However, this method leaves the choice of context to chance: determined by how the documents are organized within the training set. It would seem that a better option would be to choose the context that best represents the entities to be tagged. To that end, this work serves to provide a more useful and informative training set from which to pull context information.

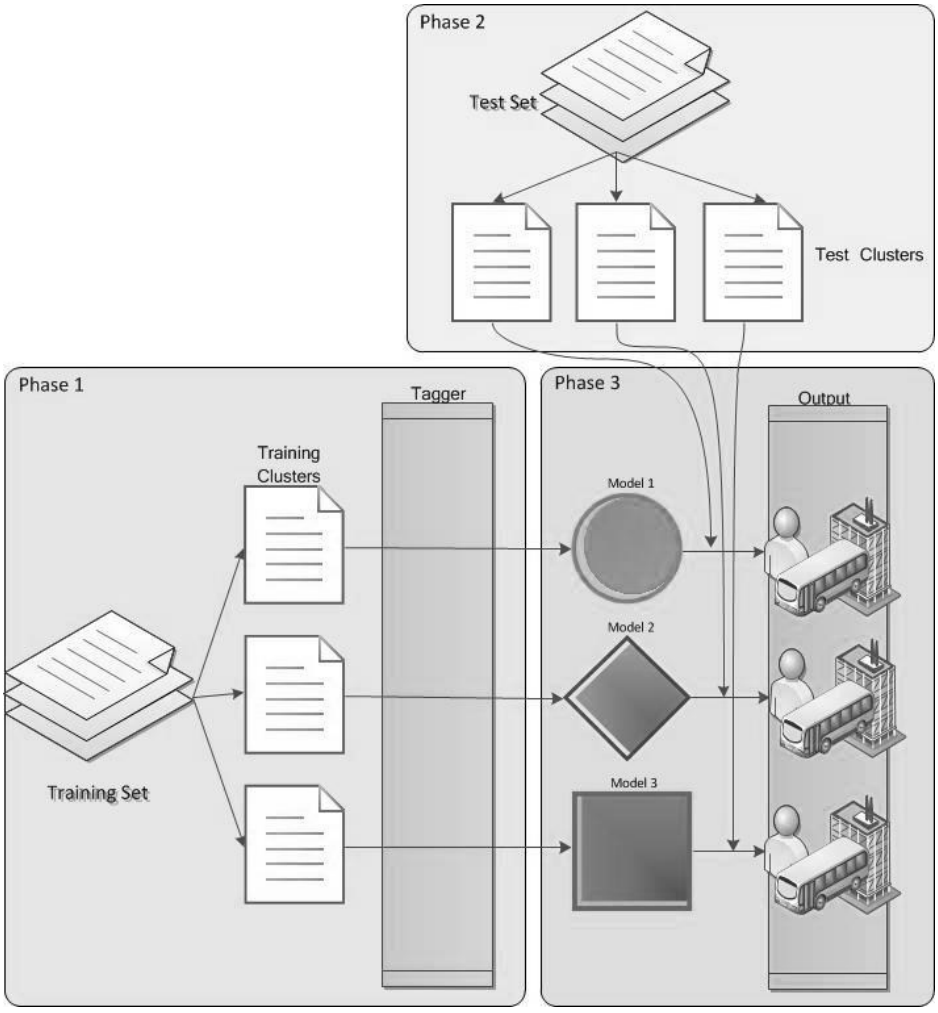

Figure 3.1 – Diagram of approach

Depicted in figure 3.1, the training set clustering technique presented in this research adapts the typical methodology utilized by standard statistical NER frameworks. For my approach, all the training documents are clustered into smaller groups based on a given similarity measure. Each of these clustered groups is then inputted to train a model using the targeted NER system. Test or input documents can then be clustered together with the training cluster that they are the most similar to and tagged using the model that was trained on that cluster. In this way, test documents are tagged with the model that is

most likely to contain contexts and features that are relevant and useful for that text. The models make predictions as to the likely classifications for the words within the test documents and named entities within the texts are identified. This process produces smaller, less noisy models that allow for increased robustness and scalability using existing NER resources.

3.1 TOP SIMILAR DOCUMENTS

The hypothesis explored in this work is that the context aggregation feature would prove more useful if the training data were more specific to the target entities. For this initial work, documents from the training set were compiled based on their similarity to the target document. These documents were then used to train a model for the LBJ tagger. In this way, I strived to reduce the noise present in the context aggregation feature as a result of the generic contexts found in a large, often heterogeneous, training set and produce feature values that are more representative of the target entities, thus producing more reliable output labels.

For an initial proof-of-concept test, for each test document, a specified number of the top documents from the training set most similar to that test document was collected. For this experiment, a simple cosine similarity measure was used. These top similar documents were used as a training set for the LBJ tagger, and the test document was then tagged using the resultant model. The system was tested by pulling the top 20, 50, 100, and 300 similar training documents to train the models. Creating training sets of larger than 300, which represents roughly a third of the entire training set, would diminish the efficacy of the experiment in trying to demonstrate that significantly smaller training sets can compete with the larger, full set. The performance of this customized model is

compared to that of the standard, two-phase LBJ tagger trained on the full CoNLL 2003 training set.

For this research, because each test document is tagged using a different model, performance was measured on a per-document basis, rather than the standard overall measure for the entire test set.[7] This performance is compared to that achieved by the standard LBJ tagger on the same document. Figure 3.2 shows the percentage of documents that were tagged more accurately using the proposed system compared to the LBJ tagger.

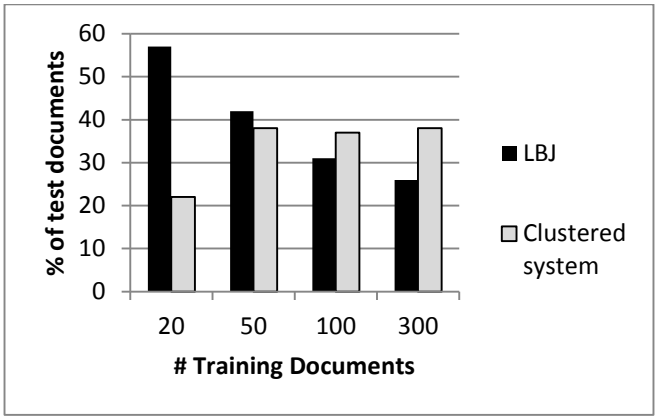

Figure 3.2 – Number of documents for which each system achieved better F1 scores.

Further, figure 3.3 displays the average percentage better and worse in terms of F1 score for each training document size. In contrast to figure 3.2, figure 3.3 demonstrates the average difference in F1 scores between the LBJ tagger trained on the

[7] The Illinois NE tagger only provides performance information in the form of percentages and does not give enough information to calculate an overall F1 score for the test set using the CoNLL eval script.

entire training set and the proposed system trained on varying numbers of training documents. These numbers indicate that there exists an optimal balance that can achieve the dual advantages of having a smaller, more relevant training set while also maintaining enough data to ensure enough features to accurately predict NER labels.

The overall aggregated difference is also provided as a more global view of performance achievements. This measurement is calculated by multiplying the F1 score of a given document by the number of entity tokens contained in that document, summing these calculations, and then dividing by the total number of entity tokens across the test dataset. Though the overall F1 score for all test documents was lower at 90.55 than the 90.77 achieved by the model trained on the entire training set, the fact that of the individual training sets achieved better accuracy for a majority of the test documents illustrates that the entire training set is not needed for effective NER tagging. Rather, a process must be established for determining which training documents are suitable for use with a given test set.

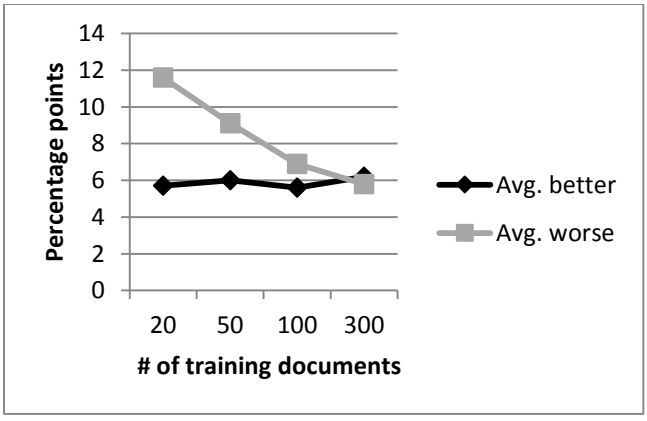

Figure 3.3 – Average percentage points better and worse in the F1 score that the proposed system achieved compared to the standard LBJ tagger for

models trained with the top 20, 50, 100, and 300 similar documents.

Because clustering a training set into groups will result in smaller training sets, it was important to first determine that performance and accuracy would not deteriorate under these conditions. These initial results demonstrate that an available training set can be easily tailored to better serve the needs of a target data set that differs from the training set and showed improvements on an existing competitive NER system by modifying the training data set used to build the prediction model. By identifying a smaller, relevant training set, the sequence tagging model is better equipped to accurately predict output labels for target data that does not closely align with the training documents.

This research has implications in the NER domain adaptation space as it demonstrates that fewer training documents are required as long as they are sufficiently similar to the targeted test set. This methodology could allow for better utilization of existing, freely-available (possibly generic) training sets by extracting portions of the training set that are more similar to the target data. It also allows for existing NER systems to be better adapted to domain-specific data without modification for feature augmentation or the inclusion of additional external data sources. Aggregating the most similar training docs for each document to be tagged is not feasible on a larger scale. Pre-clustering the original training set into smaller, more focused groups is a doable approach that allows target documents to be matched with the group of documents that is most likely to contain relevant features and example entity instances.

3.2 K-MEANS

For this application, I used the k-means implementation provided by sci-learn (Pedregosa et al., 2011). Trials were conducted with two, three and four clusters using a TF-IDF vector representation of the documents. Initially, test and training documents were all clustered together at the same time. I also experimented with clustering only the training documents and then fitting the test documents to those cluster. During all of these tests, none of the clusters performed better than the original model trained on the whole training set. Performance was sufficiently poor so as to preclude any further testing with different cluster sizes; not once, in any of the trials, did any cluster achieve a higher F1 score than the original model. Presented in Table 3.1, the highest scoring configuration involved fitting the test documents to the training documents in four clusters, two of which were extremely large and contained the majority of the test and training documents.

Table 3.1 – K-means cluster results; details F1 score of the test system, the performance of the model training on the entire training set, and the make-up of the clusters.

#	Test system	Full set	Test docs	Train docs
1	90.36	92.03	102	310
2	84.89	88.41	1	25
3	92.81	97.98	4	26
4	89.06	91.39	124	585

3.3 TOPIC MODELING

The topic modeling representation was chosen based on a desire to conduct more of a fuzzy clustering in which training documents could appear in multiple clusters. With the k-means implementation, each document can be present in only one cluster, which limits the ability for an especially useful training document to be used in multiple models and improve the tagging of a larger number of test documents. With topic modeling using Mallet, the training set is used to determine the relevant topics of a given document based on a pre-set parameter (McCallum, 2002). Based on those overall topics, each training document is then compared to each topic, with the top topics being outputted with the document's similarity to that topic.

One limitation to this approach is the inability to specify the number of desired clusters. Given each document's list of most similar topics, I experimented with a number of different criteria for assigning documents to clusters. As an initial test, I began by simply putting the document in the cluster of the topic that held the highest similarity. However, that did not achieve my goal of allowing documents to be in multiple clusters. I then provided a threshold for the similarity value, above which the document would be included in that cluster. This resulted in extremely haphazard and irregular clusters, as well as clusters that did not contain any training documents, only test documents. The ultimate configuration was to put documents in their top two topics. Using the top three topics resulted in too many clusters. I also experimented with varying the number of topics, using 7, 10, 20 and 30.

The best-performing configuration was 20 topics which resulted in four different clusters containing test documents (clusters 1, 3, 6, and 9).[8] The overall F1 score for the best-performing configuration was 90.52. Table 3.2 details the topic model performance by cluster. Though performing better than the k-means clusters, the topic model clusters were uneven and could be rather large. This defeated the purpose of trying to decrease the number of training documents in the clusters and created a larger training time for the whole system.

Table 3.2 – Topic model cluster results; details F1 score of the test system, the performance of the model training on the entire training set, and the make-up of the clusters.

#	Test system	Full set	Test docs	Train docs
1	88.547	88.245	22	461
3	0.000	66.667	1	1
6	90.586	90.175	156	903
9	93.222	93.670	52	507

3.4 TF-ICF AND COSINE SIMILARITY

For TF-ICF and cosine similarity, I chose to use an implementation of the work of Reed et al. (2004) developed at Oak Ridge National Laboratory. Most of the engineering of the system is designed for performance in creating the clusters, and it is backed by

[8] The training documents clustered into more clusters, but I only used the clusters that contained test documents.

simple concepts that can be reproduced using freely available code. When clustering the training documents, varying the similarity threshold resulted in a varying number of clusters. A similarity threshold of 0.012 yielded seven cluster groups, whereas a larger value yielded too few groups and a smaller value yielded too many. The training documents proved to be more evenly distributed in these clusters than when using the topic models.

This clustering approach provided the best accuracy of the three approaches, with F1 scores that were comparable to those achieved using the entire training set. The cluster-based models achieved an F1 score of 90.57, compared with 90.77 for the larger model. Table 3.3 gives the results of the individual clusters.

Table 3.3 – TF-ICF model cluster results; details the original test system F1 score, the F1 score after augmenting the clusters (Test+), the performance of the model training on the entire training set and the cluster make-ups after augmentation.

	Test	Test+	Full set	# Test docs	# Train docs
1	88.921	88.905	90.145	24	160
2	96.341	96.495	95.856	64	284
3	86.195	86.114	86.398	58	273
4	74.641	78.641	86.792	6	52
5	91.350	91.135	91.095	58	301
6	85.714	85.106	82.667	9	67
7	93.951	93.484	93.321	13	102

3.5 CLUSTER ADAPTATION

One adaptation was made to the clusters to address a limitation in the best-performing (TF-ICF) approach: training documents could not be included in multiple clusters. The negative impact of this fact was observed when one of the clusters ended up only containing three training documents to the six clustered test documents. When this instance arose in the test, a decision had to be made as to which documents would be added to supplement the training documents already contained in the cluster. This is also a way of remedying the problem of training documents only being assigned to one cluster.

After an analysis of the data set and the previous work in which each test document was matched with the training documents that were most similar, a number of training documents were identified as frequently occurring in these groups of most similar documents. As a result, all clusters were augmented with the 50 most similar documents (minus duplicate documents that were already contained in the cluster) to smooth out clusters and ensure that each cluster comprised an adequate number of training documents to train a model. Using the optimally performing configuration for the TF-ICF technique, performance improved from an F1 score of 90.57 to a score of 90.68 with the inclusion of the additional documents. The results of these clusters are highlighted in Table 3.3. This demonstrates that the clusters did benefit from the information contained within this universal document set without resorting to a model that includes all the training documents.

Further proposed work in this area includes examining what makes this universal set more useful than the other training documents. In particular, it would be interesting to

determine whether there are identifiable features of these documents that set them apart from the rest of the training documents. I am interested in establishing whether there is a way to identify more useful or relevant documents by only examining the training set itself or whether it is sufficient simply to extract this universal set in the same way that was used in the completed work.

Another cluster adaptation that warrants exploration is whether there are optimal cluster sizes, both training and testing, and whether performance can be improved by combining smaller clusters together. For the TF-ICF clusters, clusters 4 and 6 contained less than ten test documents each. When combined, though little change in accuracy is observed, with an F1 score of 90.69, the total training time is decreased with the deletion of one necessary model.

Chapter 4 Improving robustness and versatility

Chapter 3 establishes document clustering of training sets to be a viable method to improve the use of existing NER resources. However, to establish the robustness and broad applicability of this approach, I extend it to other taggers and training data. First, I apply the clustering approach on the Stanford tagger to determine whether the technique can be utilized with any tagger or is reliant on the underlying NER system. Similarly, I test additional data using this approach to highlight the method's flexibility to handle a broad range of available tagged data sets. Adding in more data to the training set clustering and model generation process provides for broader coverage for the NER system in general. Finally, in an effort to further increase the robustness of the NER process, an annotation tool was developed to facilitate tagging of domain-specific training data, if human resources are available. This tool incorporates a ranking algorithm that decreases the amount of data that must be tagged without a noticeable decrease in performance.

4.1 TAGGER VERSATILITY

One of the advantages to this research is that a specialized tagger is not required to implement the proposed techniques. Because the clustering does not manipulate the actual data being used to train the models, this technique does not have any bearing on the choice of tagger being employed. Furthermore, no modifications must be made to

existing systems. In this way, the approach is flexible enough to be integrated into any NER system.

Two of the most common open source NER taggers are those produced by the Cognitive Computation Group at University of Illinois at Urbana-Champaign and by the Stanford Natural Language Processing Group. Both are considered to be state-of-the-art generic taggers in the field. The experiments conducted to test the clustering techniques were originally established using the LBJ tagger from Illinois to take advantage of the feature aggregation component. To verify the transferability of the technique, the clusters created out of the CoNLL ’03 data using the TF-ICF and cosine similarity clustering technique were then run on the Stanford tagger. Transferring these experiments required only minor modifications to the format of the training data and no alterations to the Stanford tagger, demonstrating the ease with which this approach can be integrated with an existing system.

From the previous work of Ratinov and Roth, the Stanford tagger was shown to do worse in comparison to the LBJ tagger, with F1 scores of 87.04 and 90.74 respectively (Ratinov & Roth, 2009). Given this, when comparing the performance of both taggers trained on the full CoNLL training set, it was expected that the Stanford tagger would achieve slightly lower F1 scores overall on the clusters than did the LBJ tagger. The F1 scores achieved on each cluster’s test set are presented in Table 4.1.

Table 4.1 – Comparison of F1 scores between LBJ and Stanford taggers trained on full training sets

	Full LBJ	Full Stan	# Test docs
1	90.14	85.36	24
2	95.85	89.05	64
3	86.39	72.66	58
4	86.79	70.92	6
5	91.09	86.47	58
6	82.66	75.81	9
7	93.32	91.04	13

Rather than compare the F1 scores of the two taggers to each other, the experiment was designed to test the performance of the model trained on the training set from each cluster and compare that with the model trained on the entire training set. In contrast to the LBJ tagger, none of the clusters trained on the smaller training set using the Stanford tagger achieved a better F1 score than the model trained on the full training data, as demonstrated in Table 4.2. However, for the majority of the clusters, the difference in scores is not significant and would not render the tagger ineffective. Table 4.2 provides the F1 scores for the original clusters run on the Stanford tagger and those augmented by the universal set (Stan and Stan+, respectively), the scores from the clusters run on the Stanford model trained on the full training set (Full Stan), and the training and test set sizes for each cluster.

Table 4.2 – F1 scores of Stanford tagger model trained on full training set compared with cluster-based models

	Stan	Stan+	Full Stan	# Test docs	# Train docs
1	82.84	82.86	85.36	24	160
2	88.54	88.54	89.05	64	284
3	71.56	71.59	72.66	58	273
4	62.94	62.94	70.92	6	52
5	84.07	84.15	86.47	58	301
6	52.71	51.16	75.81	9	67
7	85.47	85.71	91.04	13	102

The discrepancy in performance between the taggers is likely due to the lack of context aggregation feature in the Stanford tagger. A component of the LBJ system, feature aggregation is directly impacted by document clustering as a result of the subsequent organization of the training set. The absence of this component in the Stanford tagger diminishes the effectiveness of the clustering, though it does not render it useless. Smaller, focused training sets would be less effective in the absence of such a component. The inclusion of the universal set proved to do little to improve the F1 scores. Despite the slight drop in F1 scores for the clusters as compared with training on the full training set, it is clear that the Stanford NER system can also be integrated with the training document clustering technique, thus verifying the versatility of the approach with different types of available taggers. This allows for the Stanford tagger to be made more robust by including additional training data without adding significant time and computational constraints.

4.2 AVAILABLE DATA SETS

In order to determine the limitations of this method on diverse data sets, I explore the efficacy of the approach on different sets of available tagged data. To this end, three diverse NER corpora were identified and acquired: CoNLL 2003, Ontonotes 4 and a Twitter data set. These data sets have been used in previous NER research, with results available with which to compare performance improvements. These data sets are taken from sufficiently different sources so as to represent a wide range of out-of-domain data when compared to each other.

It was the original intent to include the tagged Twitter set in the experiments to represent another out-of-domain data set in addition to Ontonotes. However, it was quickly determined that Twitter textual data would not be a good candidate for use with this technique. To conduct document similarities for clustering, documents are represented in their vector forms based on their word frequencies across the dataset lexicon. Because tweets are so short, they contain very few words relative to the entire lexicon, resulting in extremely sparse vectors. This creates difficulties when trying to cluster them using traditional document clustering techniques because the majority of tweets will have no words in common at all. These observations about the ineffectiveness of the proposed techniques on Twitter data will be incorporated as recommendations for the use of these methodologies. The Ontonotes dataset represents a broad variety of textual genres and provides adequate out-of-domain examples.

All experiments on training set clustering detailed in Chapter 3 were conducted using the CoNLL 2003 training and test data. This was done to ensure consistency across experiments. To test the viability of the technique on other data sets, the Ontonotes data

set was clustered using the best-performing clustering option – TF-ICF combined with a form of cosine similarity. After some trial-and-error experimentation on similarity threshold values, a threshold value of .30 was decided to produce adequate clusters. Since the threshold value determines the granularity of the clustering, smaller values produced a smaller number of large clusters while larger values produces a significant number of small clusters, some containing only one or two training documents. Initially the data set seemed to split into two or three large clusters with the remaining documents spreading out into a large number of much smaller clusters. Because the data set is large – 7351 documents – and in an effort to even out the clusters, the methodology for finding the universal set of documents was employed to identify the top documents in the training set most similar to the test set. A training set was aggregated using the documents that fell within the top 1000 most similar documents and had a frequency of at least 200. This resulted in a set of 2712 documents that clustered slightly more evenly. Eliminating any clusters with less than 20 training documents produced eight clusters, three of which contained significantly more documents. The results of this configuration are presented in Table 4.3.

Table 4.3 – F1 scores for Ontonotes clusters using LBJ tagger compared with model trained on entire Ontonotes data set

	Test	Full set	# Test docs	# Train docs
1	83.86	85.46	418	1283
2	30.50	46.22	154	276
3	74.14	80.45	500	842
4[9]	0	0	28	63
5	1.76	52.13	48	72
6	2.74	24.39	115	77
7	18.07	75.50	69	76
8	3.28	75.68	31	23

Though the first three larger clusters exhibited F1 scores that could prove to be usable in a real-world setting, clusters 4 through 8 experienced inferior performance due to the small size of their training sets. One way of ameliorating this problem would be to amalgamate those clusters into the other clusters[10], thus ensuring that the training data is not lost and providing the corresponding test documents with a more substantial model with which to be tagged. This rearrangement revealed a substantial improvement in cluster 2 (Table 4.4(a)).

It should be noted that only 2712 documents out of the training set were used to make the clusters and train those models though the F1 scores are compared with the model trained on the entire Ontonotes training set made up of 7351 documents. This

[9] Cluster 4 contains no entities.

[10] In this case, the clusters were combined with the smallest of the three, cluster 2, in an effort to keep training times to a minimum and improve performance on that one cluster.

information is significant when noting that the first three clusters maintained comparable. A more realistic comparison would be conducted by comparing to a model trained on only those 2712 documents, as opposed to the entire training set. Table 4.4(b) highlights the results of the three clusters compared with the model trained on only the data set composed of the frequently occurring Ontonotes documents in per-document similarity clusters. This experiment demonstrated much more comparable cluster performance compared to the model trained on the whole training set and further validates the technique as a manner of improving both robustness and scalability of a variety of available NER systems.

Table 4.4 – F1 scores for top three Ontonotes clusters with combined smaller clusters using LBJ tagger compared with model trained on (a) entire Ontonotes data set (7351 documents) and (b) smaller top Ontonotes data set

	Test	Full set	# Test docs	# Train docs
1	83.86	85.46	418	1283
2	50.34	60.71	445	587
3	74.14	80.45	500	842

(a)

	Test	Top set	# Test docs	# Train docs
1	83.86	82.07	418	1283
2	50.34	60.64	445	587
3	74.14	78.38	500	842

(b)

4.3 ANNOTATION OPTIMIZATION

At times, human resources may be available that would enable domain-specific data to be annotated for inclusion into the model-generation process. Most often, however, minimal time and labor resources are available, and an effort must be made to optimize the use of the time spent annotating text. In the event that domain-specific annotated data is desired, techniques are available to ensure that the time spent manually tagging data is spent in the most efficient manner. The research field of active learning attempts to address this problem by generating models designed to choose relevant training instances, whereby reducing the amount of training data required without impacting accuracy. However, several limitations arise within the complexity of implementing active learning that prohibit them from use by non-technical organizations. Particularly, sequence labeling tasks require more complicated algorithms to compute metrics such as diversity, representativeness, uncertainty, etc.

4.3.1 Annotation tool

An annotation tool was developed to facilitate the tagging process and decrease the amount of time spent manually annotating a training set, in the event such a training set was desired or necessary (Taylor & McKenzie, 2013). Figure 4.1 shows a snapshot of the annotation tool graphical user interface. While most active learning methods either begin with an untagged corpus or a human-annotated set of seed data, the developed approach conducts an initial tagging of the data using an open source tagger trained on the standard training set. Any freely available NER tagger could be utilized in this implementation, making the approach extremely flexible. In this way, future annotators

are required only to correct already tagged data, rather than tag plain text, thus reducing the amount of time spent on manipulating the data. Pre-tagging the plain text also serves to decrease the amount of overall work to be completed by the annotators, thus likely reducing the amount of errors introduced.

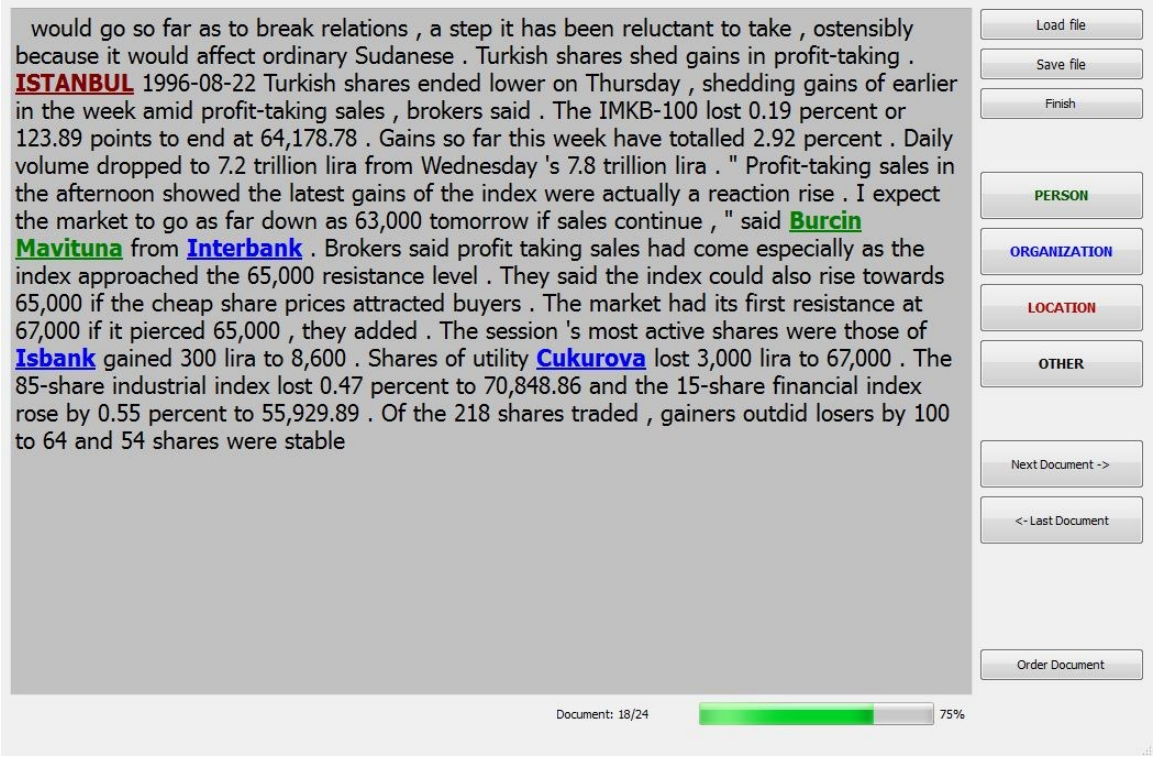

Figure 4.1 – Screenshot of tool to facilitate NER text annotation

To facilitate the tagging process, the data set is first annotated using the open-source NER tagger developed by the Stanford NLP group (Finkel, Grenager, & Manning, 2005). Once the training set has been tagged, a document ordering technique is employed to re-order the training set to ensure that the most useful parts of the data set are being corrected first, thus decreasing the amount of tagged data required. Next the data analyst can manually tag/correct the re-ordered training set using the color-coordinated

annotation scheme provided by the tool developed for this work (Taylor and McKenzie, 2013). When finished, the tool can output a correctly formatted training set based on the corrections made by the analyst. This corrected training set is then used to build a revised model to be utilized in automating the tagging of the target data. Due to its frequent use in previous NER research, the CoNLL-2003 shared task data was chosen to demonstrate the applicability of the proposed technique (Ratinov & Roth, 2009).

4.3.2 Ordering algorithm

If portions of a larger data set are chosen at random to be annotated in order to produce a NER training set, it is not guaranteed that the most useful portions of this data are being utilized, thus creating a need for more tagged data to generate a more robust model. In an effort to reduce the amount of effort expended by already time-constrained human annotators in creating training sets, an algorithm was developed to order the data set so that the most useful portions will be annotated first and less tagged data is required overall.

This approach addresses a perceived limitation in active learning techniques in that many are dependent on the machine learning algorithm being used to make the incremental sample data selections. The use of a generic function for ordering documents that does not depend on the underlying machine learning algorithm being employed means that this technique is much more adaptable for the future use of a variety of diverse NER methods and algorithms, in particular, the popular voting scheme in which a number of different models are used in combination.

To determine the best method for ordering training sets, two scoring functions were developed for ranking portions of text based on the entities that they contain. The first scoring function sorts the training set by individual sentences based on the ratio of the frequency of a given entity within the sentence to the frequency of that entity in the entire corpus.

The sentence scoring function is denoted as:

$$\left(N_{t_1} - N_{s_1}\right) + \left(N_{t_2} - N_{s_2}\right) + \cdots + \left(N_{t_x} - N_{s_x}\right)$$

where N_{t_i} is the number of occurrences of an entity within the training set and N_{s_i} is the number of occurrences of an entity within the sentence. The benefit obtained by using the sentence scoring function is that the rarest entities will be positioned at the beginning of the document. To illustrate this point, if there is only one occurrence of *Jon* in the entire training set, the sentence containing the word *Jon* will be placed at the beginning of the document. However, if *Jon* appears 50 times in the training set, a sentence containing one instance of *Jon* would not be placed as close to the beginning. This scoring function also ensures a sentence holding all or most occurrences of one entity will appear at the beginning; otherwise your final training set may not include any instances of that entity. In a real world situation, sentences will contain more than one unique, tagged entity. In this instance, each entity affects the score of that sentence, meaning the training set will be ordered by sentences with the rarest entities near the front.

An alternative scoring function was also developed that is based on 500-word blocks. The larger block size allows for a more complex function that takes into account more information about the type and uniqueness of entities within the block as compared

to the sentence-based function that only looks at entity frequencies. The 500-word scoring function is denoted as:

$$\left(\sum_{i=1}^{N_T}\left|\left(\frac{E_{t_i}}{N_e}\right)-\frac{1}{N_T}\right|\right)+\left(\frac{N_w}{N_e}\right)+\left(\frac{N_e}{N_u}\right)$$

where E_{t_i} is the number of occurrences of a specific entity type (e.g. Person, Location), N_e is the number of tagged entities, N_T is the number of types, N_w is the number of words, and N_u is the number of unique entities. The scoring function contains three components. The first is a summation of each entity type's deviation from $\frac{1}{Nt}$, which details the amount of entities of a given type that occur in that block as compared with the total number of that type across the training set. The second component is the ratio of words in the block to the number of tagged entities, which highlights how much of the block is made up of entities. The final component is the ratio of tagged entities to the number of unique entities in the block. Blocks with the highest rank are the ones that contain a significant amount of entities of a given type, a larger number entities in general, and a number of unique entities. Due to the output of the equation, blocks with the lowest score are determined to be most useful and are placed in the front of the document.

For these experiments, a training set size of 16k words was utilized. This training set was then sorted using each of the proposed scoring functions. Next a portion of the beginning of that training set was used to train the model. Finally, the model was tested on the same test data for each case. The procedures were evaluated using F1 score, a standard NER performance metric. Also, as a baseline measure, the techniques were compared against the model trained on the same training set that had not been ordered.

As seen in figure 4.2, the F1 score of the model created by the unordered set immediately drops as the size of the training set decreases. Conversely, the F1 score of the documents ordered by sentence is maintained until the number of words drops to 8000. Both scoring functions outperform the unordered data set and effectively reduce the amount of necessary tagged data by half, demonstrating that document ordering is an effective technique for reducing the burden on human annotators.

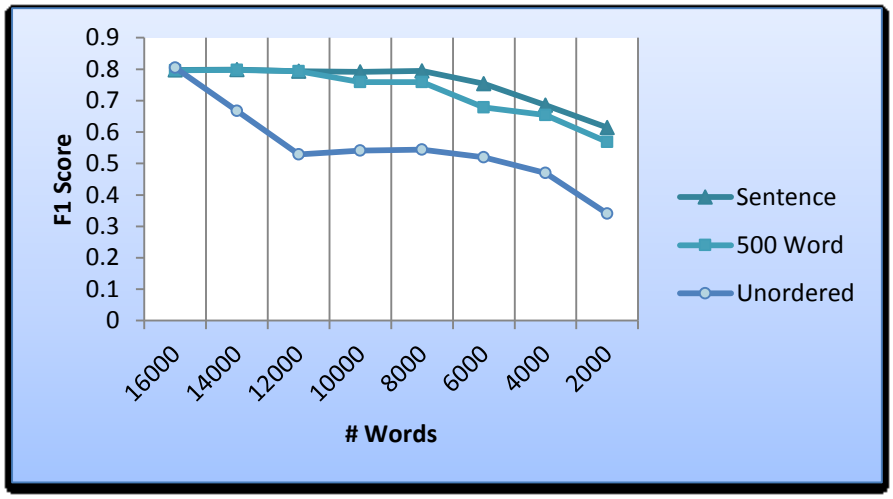

Figure 4.2 – F1 score trends using document ordering compared to unordered training sets

Chapter 5 Improving scalability

A goal for NER systems in general is to improve scalability, which means the ability to scale to larger amounts of training data. Scalability is a major motivation behind the proposed techniques of training set clustering and cluster model generation. Simply including a significant amount of training data to increase coverage is one obvious way of improving the applicability of a given NER model. However, training times increase exponentially as training sets get larger and creating a large, complex model may be time-prohibitive. One of the goals of this approach is to decrease the amount of noise and create more focused, tailored training sets, specifically to address this challenge.

5.1 CoNLL

Not only do the clusters perform comparably in terms of accuracy, but training the individual cluster models proved to take less time overall than training one large model using all of the documents from the CoNLL training set. In an attempt to simulate larger training sets, training sets were multiplied by four, and models were trained again to test training times. The training times as well as original cluster sizes are highlighted in Table 5.1. At this scale, the model trained on the training set consisting of four times the whole training set failed to complete within the default memory settings of 4GB. However, the cluster training sets, even after being increased fourfold, were able to complete within the default memory settings. These tests demonstrate the benefits of smaller training sets in terms of training times and memory requirements are observable. During all of the

proposed work, time and memory use will be recorded to determine the overall picture of how clustering training sets impacts NER system performance and to identify any areas that might need alterations or improvements based on observations.

Table 5.1 – Performance of clustering technique compared to training a model using the full training set for both the original training documents and doubled training sets.

	# docs	Training time	4x
Cluster 1	160	1m46s	3m37s
Cluster 2	284	2m13s	4m39s
Cluster 3	273	2m3s	3m58s
Cluster 4	52	1m6s	1m49s
Cluster 5	301	2m18s	4m45s
Cluster 6	67	56s	1m49s
Cluster 7	102	1m9s	2m28s
Total time		**11m31s**	**23m5s**
Full data set	**946**	**14m13s**	**N/A**

The technique can be extended to any NER tagger without tagger modification due to its manipulation of the input data rather than the tagger itself. This is demonstrated by conducting the same experiments on an additional open-source tagger, that produced by the Stanford NLP group. The Stanford tagger maintains higher memory requirements than the Illinois tagger because of its underlying machine learning framework. For this reason, memory settings were raised to ensure of the completion of some model training, though not to levels that might not be achievable by the average forensic investigator.

The results of these experiments mirrored those observed with the Illinois tagger. The combined training times for the cluster-based models did not approach that of the one model training time for the combined training set. These results are presented in Table 5.2.

Table 5.2 – Clustering technique compared to training a model using the full CoNLL training set for both the original training documents and quadrupled training sets using Stanford tagger.

	# docs	Training time	4x
Cluster 1	160	1m16s	4m24s
Cluster 2	284	2m59s	7m37s
Cluster 3	273	3m54s	10m28s
Cluster 4	52	0m35s	1m58s
Cluster 5	301	2m21s	6m58s
Cluster 6	67	0m36s	2m12s
Cluster 7	102	0m49s	2m42s
Total time		**12m30s**	**36m19s**
Full data set	**946**	**14m58s**	**60m21s**

Though most NER research focuses on F1 score as a measure of the success of a given NER system or technique, other measures exist for determining the outcome of this type of research. Training time is one and has already been addressed for this approach. In addition, accuracy, precision and recall are other common measurement statistics, though precision and recall are already included in the calculation for F1. Given that the goal of this system is to ease the burden and complexity of NER for resource-constrained organizations, it is also necessary to ensure that the proposed technique accomplishes these goals by measuring savings of time and resources, as has been noted previously. These measurements will be taken into account for the final presentation of this research.

5.2 CoNLL and Ontonotes

Further experimentation combining the CoNLL and Ontonotes data sets revealed similar performance gains when conducted on the Illinois tagger. Due to the variety of different types of data contained within the data set, the inclusions of Ontonotes broadens the applicability of the training set to a number of other domains and widens the coverage

of the model. This serves to facilitate domain adaptation and the efficacy of exiting NER taggers and data sets, while bypassing the need for manual tagging. However, adding in the Ontonotes data makes for a significantly more complex model due to a larger part-of-speech and named entity tagsets, resulting in much longer training times. While the cluster-based models could be trained using the pre-specified 4GB of memory, the large combined model ran out of memory after 3 hours and 20 minutes, necessitating an increase in memory to 8GB. Despite the superior memory allotment, the model trained on the full CoNLL and Ontonotes data sets nevertheless took a significantly longer amount of time to train than the combined training times of the models trained on the clusters, as illustrated in Table 5.3.

Table 5.3 – Clustering technique compared to training a model using the full CoNLL and Ontonotes training sets using Illinois tagger

	# docs	Training time
Cluster 1	1160	50m1s
Cluster 2	1284	69m6s
Cluster 3	1273	65m46s
Cluster 4	1052	23m33s
Cluster 5	1301	82m33s
Cluster 6	1067	17m17s
Cluster 7	1102	33m10s
Total time		**5h41m38s**
Full data set	**7946**	**24h13m46s**

For the combined CoNLL and Ontonotes training set, the memory requirements of the Stanford tagger exceeded the previously used settings of 4GB and 8GB and would not successfully complete. This is largely due to the significant amount of features generated for CRF model development. To overcome this limitation, only the word and

its NER tag were passed in the training stage, and the part-of-speech and chunk tags were discarded. This allowed for the cluster-based models to be trained. However, the full model trained on the whole training set was unable to complete as the system ran out of memory trying to generate the model.

Table 5.4 – Clustering technique compared to training a model using the full CoNLL and Ontonotes training sets using Stanford tagger

	# docs	Training time
Cluster 1	1160	62m17s
Cluster 2	1284	74m35s
Cluster 3	1273	75m5s
Cluster 4	1052	28m9s
Cluster 5	1301	99m45s
Cluster 6	1067	20m12s
Cluster 7	1102	40m56s
Total time		**6h50m59s**
Full data set	**7946**	**N/A**

The approach presented in this work demonstrated that clustering the training set into more focused smaller groups allows more data to be incorporated into the training process, whereby increasing the efficacy of existing taggers and tagged data sets and avoiding the necessity to manually tag training data. It was shown that smaller clusters can be trained in less time and using less memory than one larger cluster using all the training data.

Chapter 6 NER recommendations

This research can be taken a step further by expanding the approach into a set of recommendations for organizations, in particular, law enforcement, as to the best options for implementation for their data. Data contains general characteristics that will lend itself well to being tagged with more generic approaches or makes it necessary to employ modifications such as are presented in this technique. It is important for law enforcement analysts to have an idea as to when the use of this approach is required or recommended.

The development of this research revealed certain aspects of the methodologies that lend themselves to a certain usage or optimization. One of the goals of this research is to provide an approach that is more accessible to non-scientific organizations. To establish a complete solution, recommendations must be offered as to the use and optimization of the approach. These recommendations serve to eliminate much of the guesswork involved in the implementations of these ideas in a real-world setting and to ensure that the best possible performance is achieved on the NER tagging task. In general recommendations can be made in the areas of:

- Data
- Clustering
- Taggers
- Performance

6.1 DATA

Choice of training data is the first consideration that must be taken into account during implementation of any NER system. For this approach, the structure of the training set is impacted by the document clustering and must also be taken into consideration. Most document clustering methods rely on vector-based similarity measures. These document vector representations impose constraints on the types of documents that can be effectively clustered to make the focused training sets. When the document length is too small, vectors become too sparse, making it difficult to compare for similarity and therefore create clusters based on that data. For that reason, this technique would not perform well on data that naturally contains documents with short text fragments – such as Twitter or individual chat messages. It is recommended that Twitter, or other similar, data be analyzed separately when utilizing the described approach.

Though not appropriate for Twitter, the proposed technique does facilitate the use of multiple disparate data sets. Generally, combining a number of significantly different data sets into one training set could potentially generate an extremely noisy and inaccurate model, thus negatively impacting performance. By employing the clustering technique, data sets from significantly different domains, possibly with their own tag sets, will likely cluster together and wind up in different models. The initial clustering step provides a means of determining to which model each test document should be fed by way of a similarity comparison with each cluster center. Simply creating a model with each disparate data set provides no such means. In this way, organizations are free to

aggregate any freely-available and/or customized training sets in an effort to create broader coverage for the system.

For larger data sets, if the documents seem to cluster together into only a couple of similar large groups, it was found to be useful to first employ the technique to identify the universal set of documents used to augment the clusters (detailed in Section 0). The smaller training set, constructed out of an original set that had many similar documents, clusters more easily and is guaranteed to contain documents that are most similar to the target data.

6.2 CLUSTERING

During the course of the research on document clustering algorithms for NER, several best-practice recommendations were highlighted. First, it is important to note that training set clustering is independent of the NER tagger in terms of implementation. This means that any decisions affecting clustering do not need to take tagger choice into account. After an analysis of several varied document clustering algorithms, cosine similarity using TF-ICF vector representations proved to achieve superior performance over other methods tested. This technique should be included as part of the implemented NER system for optimal system configuration.

During the clustering experimentation, cluster structure was analyzed for its efficacy in the approach. Based on these observations, organizations should strive to obtain a handful of clusters that are relatively evenly distributed in terms of size. Too many clusters results in models with inadequate information; too few increases training times and introduces additional noise into the models. Some data sets may require some minor experimentation with similarity thresholds to identify the best cluster

configuration. Testing also showed that model performance degrades quickly when training sets begin to get too small – in general, smaller than 150-200 documents. Smaller clusters that would otherwise not perform well could be re-clustered into larger existing clusters to insure against a drop in performance.

Once a training set has been split into clusters, it might be tempting to train models only on clusters that contain test documents in an effort to reduce training times. However, this effectively diminishes the robustness of the system, as some previously unseen input documents might be best tagged by a model trained on one of those clusters. It is therefore recommended that models be generated for all training clusters, regardless of whether they originally contained test documents. This ensures the maximum accuracy for future input documents.

6.3 TAGGER

As was previously noted, document clustering in this approach is independent of the underlying tagger and can be employed with any available NER tagging system. This is due to the fact that clustering simply results in smaller training sets and conducts no data manipulation that would otherwise affect tagger usage. That being said, performance of the technique was found to be optimal with the NER tagger developed by the University of Illinois at Urbana-Champaign rather than that produced by Stanford's NLP group. This is likely due to the fact that the Stanford tagger has no feature aggregation component to be positively impacted by the clustering.

Though only tested on these two state-of-the-art open source taggers, the approach does not require the use of one of these taggers. Organizations should identify a tagger that is easily accessible and involves a minimal amount of complexity in terms of

implementation and configuration. If possible, it would be advantageous to employ a tagger that includes some manner of context aggregation feature that would benefit from the training set clustering involved in this research.

6.4 PERFORMANCE AND SCALING

Two of the main benefits of this approach related to performance are decreases in model training times and the scalability that results from deconstructing the training data set into smaller, focused clusters. For smaller training sets, a single model may be best rather than employing the clustering technique. This approach is appropriate for larger training sets or when adding in additional instances. Particularly for law enforcement, when the target data can be so varied, it is essential to be able to add in more tagged data to make the model more robust. If a small amount of data were to be tagged to supplement each new case, with traditional methods, only the case-specific data could be included with the general training set due to computational constraints when training a model as training sets grow. To counteract the extra complexity and training time introduced by the added data, the smaller clusters can train a number of models in less time than it would take to train one large model on all aggregated data. This technology makes it feasible to combine tagged sets from multiple cases over time, resulting in an increasingly more accurate and robust NER system.

The approach developed for this research is designed to enable organizations to more easily implement and scale NER tagging systems. By providing recommendations as to the best-use practices for the approach, the intent is to ensure that the maximum benefits in terms of accuracy, ease-of-use, and performance are realized.

Chapter 7 Conclusion

As information grows exponentially, so does the desire to analyze and make use of this data in a systematic manner. Turning to the NLP community for help, organizations attempt to utilize existing tools, often with limited success due to the difference between their data and the data on which the systems have been trained. Rather than develop specialized NER systems or ways of automatically generating new tagged data, it is imperative that methods be developed for adapting these systems for improved performance with existing systems and data.

For law enforcement or other organizations needing to conduct text-based data analysis, the implementation of state-of-the-art NER techniques can prove prohibitively complex and time consuming. On the other hand, utilizing open-source solutions often results in sub-par performance due to variations inherent in the target data and difficulties in scaling to accommodate more diverse training data. These challenges motivated this research to develop an NER approach that facilitates the use of open-source or available resources, bypasses the need for a specialized NLP expert, and allows a system to scale up to larger training sets without the need for sophisticated computational hardware.

All facets of this research were designed to address the aforementioned goals. NER was combined with document clustering – after experimentation with different clustering methodologies – to produce a novel way of computing models from training data that reduces noise inherent in larger models. Not only is this technique able to be

integrated with any NER system, it also generates smaller training sets, allowing for more tagged data to be used than when only working from a single training set. An annotation tool was also developed to simplify the tagging process for analysts not skilled in NER. This tool incorporates a method for ordering the training data which reduces the amount of data needed to maintain accuracy. Finally, recommendations were provided for the use and implementation of these tools and techniques to ensure optimal performance.

The TF-ICF and cosine similarity clustering method produced a reasonable number of clusters, and the models trained on those clusters were shown to be nearly as accurate as a single model trained on the entire CoNLL training set. In addition, the clustered training sets proved to significantly decrease training times – up to 5x speed up – as compared to training the larger model. This introduces a novel way of scaling to larger training sets and incorporating more training data, thus creating a more robust system. The document ordering technique incorporated into the annotation tool further decreases time involved in the training process by effectively cutting in half the amount of data required to be tagged while maintaining accuracy. All developed tools and techniques are independent of the underlying NER resources and can be integrated with any available tagger or data sets.

The motivation for this research stems from a need to improve the performance of available NER taggers combined with existing tagged data and leverage these resources in a way that make them more accessible and useful for organizations such as law enforcement. The contributions of this research include:

- demonstrating that smaller, more focused training sets can compete with a larger, more generic training set,

- presenting document clustering of training sets as a way of grouping together like features, whereby achieving better accuracy for out-of-domain data,
- analyzing a variety of document clustering techniques for their utility in an NER application,
- highlighting the utility of document clustering with real-world NER taggers and tagged data sets,
- demonstrating that document clustering of training sets reduces model training time and memory requirements and eliminates the need for manual tagging or system development,
- providing a tool for simplified annotation that results in less training data being required for comparable performance, and
- detailing a set of recommendations for the implementation of this approach and ways to optimize performance.

7.1 FUTURE WORK

There are several avenues of research that could be continued to improve this approach and further the research. One interesting pursuit would be an in-depth examination of the impact that clustering has on the context aggregation feature. The assumption is that clustering the training set so that similar documents are grouped together would mean that the context for a given entity would then be aggregated across similar documents, rather than across a random assortment. In theory, this should serve to provide a more representative context and improve the model. However, this hypothesis has not been thoroughly tested. In order to verify the impact that clustering has on this feature, an instance of an entity would have to be manufactured so that certain documents

contained representative contexts, while others did not. In this way, the placement of those representative documents could be tracked and the effect of clustering could be determined.

Similarly, there is also a need to be able to characterize documents and identify the usefulness they might have for the performance of the NER system as training data instances. In this way, the methodology for augmenting the clusters could be refined to take into account the actual make-up of the clusters and which documents would serve to provide the most useful information. Useful information might include the number and type of entities, document length, sentence lengths, type of document, topic clusters, etc. This document characterization could also be employed to evaluated document representation schemes specifically for NER and determine what type of representation would best serve the needs of the NER task. While much research has been conducted on evaluating NER systems and their performance for a given task or domain, no work to-date has been done to validate the usefulness of annotated data sets themselves or to determine which available data sets, or subsets of that data, would yield the best accuracy for a given target domain. This information would be extremely useful for organizations looking to make efficient use of data that has already been tagged, rather than annotating domain-specific target data.

Further work on the annotation tool would involve developing a more sophisticated document ranking algorithm using machine learning. The goal would be to develop a ranking function that did not depend on the underlying machine learning algorithm. In this way, the function itself could be optimized by integrating supervised or semi-supervised learning techniques that would be better equipped to determine the best

documents to be tagged. However, the supplied seeding instances would not have to be tailored to the particular machine learning algorithm. Work in this area could also prove to be innovative in the field of active learning, as well as NER.

References

[1] Ah-Pine, J., & Jacquet, G. (2009). Clique-based clustering for improved named entity recognition systems. *Proceedings of the 12th Conference of the European Chapter of the ACL*, (pp. 51-59). Athens, Greece.

[2] Allison, B., Guthrie, D., & Guthrie, L. (2006). Another look at the data sparsity problem. *TSD 2006, LNAI 4188* (pp. 327-334). Springer-Verlag.

[3] Becker, M., & Osborne, M. (2005). A two-stage method for active learning of statistical grammars. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence* (pp. 991-996). Edinburgh, Scotland: Professional Book Center.

[4] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Mach Learn, 79*, 151-175.

[5] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit.* Sebastopol, CA: O'Reilly Media.

[6] Brown, P. F., deSouza, P. V., Mercer, R. L., Della Pietra, V. J., & Lai, J. C. (1992, December). Class-based n-gram models of natural language. *Computational Linguistics, 18*(4), 467-479.

[7] Chieu, H. L., & Ng, H. T. (2003). Named entity recognition with a maximum entropy approach. *Proceedings CoNLL 2003*, (pp. 160-163). Edmonton, Canada.

[8] Ciaramita, M., & Altun, Y. (2005). Named-entity recognition in novel domains with external lexical knowledge. *Advances in Structured Learning for Text and Speech Processing Workshop.*

[9] Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. *Proceedings of the 25th International Conference on Machine Learning* (pp. 160-167). Helsinki, Finland: Association for Computing Machinery.

[10] Cunningham, H., Wilks, Y., & Gaizauskas, R. J. (1996). GATE: a general architecture for text engineering. *COLING '96: Proceedings of the 16th conference on Computational Linguistics. 2.* Association for Computational Linguistics.

[11] Dalton, J., Allan, J., & Smith, D. A. (2011). Passage retrieval for incorporating global evidence in sequence labeling. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (pp. 355-364). Glasgow, UK: ACM.

[12] Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, (pp. 363-370).

[13] Gantz, J., & Reinsel, D. (2011). *Extracting value from chaos.* EMC Corporation. Retrieved from http://www.emc.com/digital_universe

[14] Goldberg, Y., Tsarfaty, R., Adler, M., & Elhadad, M. (2009). Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and EM-HMM-based lexical probabilities. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 327-335). Sydney, Australia: ACL.

[15] Griffiths, T. L., Steyvers, M., Blei, D. M., & Tenenbaum, J. B. (2005). Integrating topics and syntax. *Advances in Neural Information Processing Systems, 17*.

[16] Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., & Su, Z. (2009). Domain adaptation with latent semantic association for named entity recognition. *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, 281-289.

[17] Hachey, B., Alex, B., & Becker, M. (2005). Investigating the effects of selective sampling on the annotation task. *Proceedings of the Ninth Conference on Computational Natural Language Learning* (pp. 144-151). Ann Arbor, Michigan: ACL.

[18] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: the 90% solution. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL* (pp. 57-60). New York: ACL.

[19] Huang, F., & Yates, A. (2009). Distributional representations for handling sparsity in supervised sequence-labeling. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 1*, pp. 495-503. Suntec, Singapore: ACL.

[20] Janik, M., & Kochut, K. J. (2008). Wikipedia in action: ontological knowledge in text categorization. *Proceedings of the 2008 IEEE International Conference on Semantic Computing* (pp. 268-275). Santa Clara, CA: IEEE Computer Society.

[21] Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing.* Upper Saddle River, New Jersey: Pearson Education, Inc.

[22] Kim, S., Song, Y., Kim, K., Cha, J.-W., & Lee, G. G. (2006). MMR-based active machine learning for bio named entity recognition. *Proceedings of the Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Anual Meeting* (pp. 69-72). New York, New York: ACL.

[23] Koo, T., Carreras, X., & Collins, M. J. (2008). Simple semi-supervised dependency parsing. *Proceedings of ACL*, (pp. 595-603).

[24] Krishnan, V., & Manning, C. D. (2006). An effective two-stage model for exploiting non-local dependencies in named entity recognition. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, (pp. 1121-1128). Sydney, Australia.

[25] Li, W., & McCallum, A. (2005). Semi-supervised sequence modeling with syntactic topic models. *Proceedings of the 20th National Conference on Artificial Intelligence. 2*, pp. 813-818. Pittsburgh, PA: AAAI Press.

[26] Lin, D., & Wu, X. (2009). Phrase clustering for discriminative learning. *ACL'09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (pp. 1030-1038). Suntec, Singapore: Association for Computational Linguistics.

[27] Lin, W.-P., Snover, M., & Ji, H. (2011). Unsupervised language-independent name translation mining from Wikipedia Infoboxes. *Proceedings of Conference on Empirical*

Methods in Natural Language Processing (pp. 43-52). Edinburgh, Scotland: Association for Computational Linguistics.

[28] Liu, X., Zhang, S., Wei, F., & Zhou, M. (2011). Recognizing named entities in tweets. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 1*, pp. 359-367. Portland, OR: Association for Computational Linguistics.

[29] McCallum, A. K. (2002). Retrieved from MALLET: A Machine Learning for Language Toolkit: http://mallet.cs.umass.edu

[30] McKenzie, A. (2013). Focused training sets to reduce noise in NER feature models. *The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (to appear).* Atlanta, GA.

[31] McKenzie, A. T., Matthews, M., Goodman, N., & Bayoumi, A. (2010). Information extraction from helicopter maintenace records as a springboard for the future of maintenance text analysis. *The Twenty Third International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA-AIE 2010).* Cordoba, Spain.

[32] Miller, S., Guinness, J., & Zamanian, A. (2004). Name tagging with word clusters and discriminative training. *Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, (pp. 337-342). Boston, MA.

[33] Munkhdalai, T., Li, M., Kim, T., Namsrai, O.-E., Jeong, S.-p., Shin, J., & Ryu, K. H. (2012). Bio named entity recognition based on co-training algorithm. *26th International Confrence on Advanced Information Networking and Applications Workshops*, (pp. 857-862).

[34] Munro, R., & Manning, C. (2012). Accurage unsupervised joint named-entity extraction from unaligned parallel text. *The 4th Named Entities Workshop.* Jeju, Korea.

[35] Nadeau, D., & Sekine, S. (2009). A survey of named entity recognition and classification. *Named Entities: Recognition, Classification and Use*, 3-28.

[36] Olsson, F. (2008). *Bootstrapping named entity recognition by means of active machine learning.* University of Gothenburg.

[37] Olsson, F. (2009). *A literature survey of active machine learning in the context of natural language processing.* Swedish Institute of Computer Science.

[38] Pedregosa et al. (Ed.). (2011). Scikit-learn: Machine learning in python. *JMLR, 12*, 2825-2830.

[39] Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. *CoNLL '09 Proceedings of the Thirteenth Conference on Computational Natural Language Learning* (pp. 147-155). Boulder, CO: Association for Computational Linguistics.

[40] Reed, J. W., Jiao, Y., Potok, T. E., Klump, B. A., Elmore, M. T., & Hurson, A. R. (2006). TF-ICF: a new term weighting scheme for clustering dynamic data streams. *Proceedings of the 5th International Conference on Machine Learning and Applications*, (pp. 258-263). Orlando, FL.

[41] Reed, J. W., Potok, T. E., & Patton, R. M. (2004). A multi-agent system for distributed cluster analysis. *Proceedings of the Third International Workshop on Software*

Engineering for Large-Scale Multi-Agent Systems, (pp. 152-155). Edinburgh, Scotland, UK.

[42] Ritter, A., Clark, S., Mausam, & Etzioni, O. (2011). Named entity recognition in tweets: an experimental study. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 1524-1534). Edinburgh, Scotland, UK: ACL.

[43] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation, 60*(5), 503-520.

[44] Rüd, S., Ciaramita, M., Müller, J., & Schütze, H. (2011). Piggyback: using search engines for robust cross-domain named entity recognition. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 1*, pp. 965-975. Portland, OR: Association for Computational Linguistics.

[45] Settles, B. (2009). *Active learning literature survey.* University of Wisconsin-Madison.

[46] Shah, N., & Mahajan, S. (2012). Document clustering: a detailed review. *International Journal of Applied Information Systems, 4*(5), 30-38.

[47] Shen, D., Zhang, J., Su, J., Zhou, G., & Tan, C.-L. (2004). Multi-criteria-based active learning for named entity recognition. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 589-596). Barcelona, Spain: ACL.

[48] Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *6th ACM SIGKDD, World Text Mining Conference.* Boston, MA.

[49] Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, & W. Kintsch, *Latent Semantic Analysis: A Road to Meaning.* Laurence Erlbaum.

[50] Sun, A., & Grishman, R. (2011). Cross-domain bootstrapping for named entity recognition. *Proceedings of SIGIR 2011 Workshop on Entity-Oriented Search.*

[51] Szarvas, G., Farkas, R., & Ormándi, R. (2007). Improving a state-of-the-art named entity recognition system using the world wide web. *Advances in Data Mining. Theoretical Aspects and Applications.*, (pp. 163-172).

[52] Taylor, Z., & McKenzie, A. (2013). Facilitation of supervised NER model training by document ordering. *CDAW 2013: Computational Data Analytics Workshop.* Oak Ridge, TN.

[53] Tishby, N., Pereira, F. C., & Bailek, W. (1999). The information bottleneck method. *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, (pp. 368-377).

[54] Tkachenko, M., & Simanovsky, A. (2012). Named entity recognition: exploring features. *Proceedings of KONVENS 2012*, (pp. 118-127). Vienna.

[55] Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 384-394). Uppsala, Sweden: Association for Computational Linguistics.

[56] Turian, J., Ratinov, L., Bengio, Y., & Roth, D. (2009). A preliminary evaluation of word representations for named-entity recognition. *NIPS Workshop on Grammar Induction, Representation of Language and Language Learning.* Vancouver, BC.

[57] Urbansky, D., Thom, J. A., Schuster, D., & Schill, A. (2011). Training a named entity recognizer on the web. In A. Bouguettaya, M. Hauswirth, & L. Liu, *Lecture Notes in Computer Science - Web Information System Engineering* (Vol. 6997, pp. 87-100).

[58] Usami, Y., Cho, H.-C., Okazaki, N., & Tsujii, J. (2011). Automatic acquisition of huge training data for bio-medical named entity recognition. *Proceedings of BioNLP 2011 Workshop* (pp. 65-73). Portland, OR: Association for Computational Linguistics.

[59] Ushioda, A. (1996). Hierarchical clustering of words. *COLING '96 Proceedings of the 16th conference on Computational linguistics. 2*, pp. 1159-1162. Copenhagen, Denmark: Association of Computational Linguistics.

[60] Uszkoreit, J., & Brants, T. (2008). Distributed word clustering for large scale class-based language modeling in machine translation. *Proceedings of ACL-08: HLT*, (pp. 755-762). Columbus, OH.

[61] Vlachos, A. (2006). Active annotation. *Proceedings of the Workshop on Adaptive Text Extraction and Mining* (pp. 64-71). Trento, Italy: ACL.

[62] Wu, D., Lee, W. S., Ye, N., & Leong, H. (2009). Domain adaptive bootstrapping for named entity recognition. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (pp. 1523-1532). Singapore: ACL.

[63] Zhang, T., & Johnson, D. (2003). A robust risk minimization based named entity recognition system. *CONLL '03 Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003. 4*, pp. 204-207. Edmonton, Canada: Association for Computational Linguistics.