Named Entity Recognition: Adapting to Microblogging

Senior Thesis, Spring 2009

Brian Locke
Student
lockebw@colorado.edu

Dr. James Martin, Ph.D
Advisor
James.Martin@colorado.edu

Named Entity Recognition, Adapting to Microblogging

1 Overview

In this project, we seek to create a Named Entity Recognizer (NER) tuned for use on Twitter posts. We will be identifying Named Entities and classifying them as People, Locations, or Organizations. We hope to identify language features and methods that effectively transfer the techniques and knowledge from Named Entity Recognition research on formal sources, such as news articles, to less structured microblogging texts. In the process, we will identify differences between microblogging text and formal prose which are relevant to NER.

1.1 Summary

There has been much research in Named Entity Recognition on news articles. However, many applications of NER, and Natural Language Processing in general, involve analyzing data that is less structured, such as blog posts, instant messages, and movie reviews. In our project, we will attempt to create a classifier that performs NER on microblog postings. In doing so, we hope to explore approaches to transferring learning from a domain with more data available to one with less.

2 Related and Previous Work

The previous research relating to our project can be grouped into research about Named Entity Recognition on formal sources and research about transfer learning (not necessarily related to NER). Some work has been done into domain adaptation in NER, specifically from News Texts to Bio-medical texts.

2.1 NER

The Named Entity Recognition task is concerned with marking occurrences of a specific object being mentioned. These mentions are then classified into a set of predefined categories. Standard categories include "person", "location", "geo-political organization", "facility", "organization", and "time". For example, in the sentence "The George Washington Bridge spans the Hudson River,", the phrase "George Washington Bridge" would be marked and classified as a facility, and "Hudson River" would be marked as a location. However, the sentence "The bridge spans the river" does not contain any named entity mentions, as the task is traditionally defined, because "bridge" and "river" do not refer to specific entities in this context; they are generic mentions.[11]

Researchers have explored domain specific, multi-lingual, and standard Named Entity Recognition. Several conferences have been devoted to Mention Detection: MUC, ACE, and CoNLL-2002 and 2003.[6][16][7] In these conferences, systems have been developed that achieve F-measure scores (a weighted average of precision and accuracy) of up to 88.7 on English newswire texts.[16]

2.2 Transfer Learning

In addition to the standard NER task, transfer learning is also relevant to this project. In transfer learning a classifier is trained on data coming from one distribution, called the source domain, with the intent of using that classifier on another domain, called the target domain. Transfer learning is especially difficult when the target domain and the source domain are very different. A good transfer learning algorithm identifies features which generalize, or can be adapted, between different domains.

Arnold, Nallapati, and Cohen investigated transfer learning in NER between news and biological texts, and also between different types of biological texts.[1] They attempted to use a hierarchy of features to identify the features which generalized well between the domains. Additionally, they analyzed the frequency of candidate words occurring in different domains to identify words that could be reliably classified.

In 2005, Aue and Gamon attempted to apply transfer learning to sentiment classification.[2] They investigated four different methods for transfer learning: Training on a mixture of labeled data from both domains, training on labeled data from both domains using only features observed in both sets of data, using ensemble learning with classifiers from each domain, and using a small amount of labeled data and a large amount of unlabeled data from

the target domain. They found that using a small amount of labeled data with a large amount of unlabeled data gave the best results. Their domains were movie reviews, book reviews, product support, and online survey text.

Drezde et al. also analyzed transfer learning in sentiment classification.[3] They analyzed the frequency of features between domains in order to select a "pivot set" of features that occur with similar frequency. Additionally, they devised a metric for domain-relatedness. Their measure attempts to quantify only the differences between domains that will cause performance to decrease when transfer learning is applied. They select a feature set that differs most in taxi-cab distance (as apposed to any other norm) and then calculate and estimate of the distance by finding the empirical risk in classifying between the two feature sets.

3 Data

The data that we are primarily concerned with (our target domain) will be text taken from Twitter feeds. Twitter is a microblogging utility that lets people post short updates about their life. Twitter posts have a 140 character limit, so the language structure that Twitterers use is often much different from standard text. The character limit often forces people to use abbreviations. Additionally, Twitter posts often do not contain proper capitalization (analysis shows that only 55 percent of posts start with a capital letter), contain links as the subject to their posts, and do not contain grammatically correct sentences. Despite these difficulties, the character limit may make patterns of structure, such as part of speech tags, easier for a classifier to memorize.

3.1 CoNLL-2003

The CoNLL-2003 named entity dataset will be used as our source domain in training the classifier. The CoNLL-2003 Dataset is a multilingual corpus for Named Entity Recognition compiled for the Conference on Natural Language Learning shared task in 2003. The CoNLL dataset contains articles in English and German; we are only concerned with the English portion. The English part is derived from Reuters news snippets. It is divided into a training section, developer's test set, and an evaluation test set. The training set, which contains 203,621 tokens and 23,499 entity mentions, is used to train the Support Vector Machine. The developer's test set was used by shared task participants to test their systems as they developed them. It contains 51,362 tokens and 4,942 entity mentions. There is an additional test set used by the conference organizers to evaluate an entry's performance. This test set contains 46,435 tokens and 5,648 entity mentions.[16]

The CoNLL dataset uses a tag set consisting of 4 tags: PER, ORG, LOC, and MISC. The PER tag is designated to mentions of a person by their name or by an abbreviation of their name. For example, "John Smith" is marked "PER", as is "John", "John S.", and "J. Smith". LOC is given to regions, structures, natural locations, public places, commercial places, buildings, and abstract places. ORG is given to companies, political movements, government bodies, and other collections of people. MISC is given to adjectives derived from Named Entities, religions, nationalities, events, titles, and non-brand types.[11] Part of speech and noun phrase chunking data is included with the data.

3.2 Twitter Text

Several different types of Twitter posts, often called 'Tweets', were collected. 600 Tweets were taken from the 'Public Timeline' on February 10th. The 'Public Timeline' is a collection of the latest Tweets of any type, from any user. The Tweets from the 'Public Timeline' vary greatly in structure and subject matter. These Tweets were gathered to create a body of text that reflects the general structure of Twitter postings.

Due to the varied nature of Tweets, many of the expected uses for a named entity recognition system are concerned with Twitter postings about specific events. Therefore, it was decided that some of the Tweets in our evaluation set should be taken from results about specific events. Tweets about events of 3 different types were collected. On February 10th, 200 tweets were gathered relating to the current economic recession, an ongoing, ubiquitous event. Also on February 10th, 300 Tweets relating to the Australian Bushfires, a shorter and largely regional event, were gathered. Lastly, on March 5th, 584 Tweets regarding the gas explosion in Bozeman, MT, a very quick, local event, were gathered. The topic-related posts had a much higher rate of named entity mentions. Many of the 'Public Timeline' Tweets were about feelings or mundane status updates, whereas the "topic-related posts generally focused on events which were described in terms of named entities. Both types of tweets had a

much lower rate of entity mentions (an average of 1 mention per 17.4 words) than newswire text (1 mention for every 8.7 words).

The Tweets were gathered from many different sources in an attempt to avoid evaluation anomalies. If all the Tweets that were collected related to the same subject, it would be expected that these Tweets would contain many occurrences of a few words. In that case, if the classifier got that word wrong in every case, it would have an evaluation score that was much lower than the actual performance of that classifier on a general set of text. Additionally, if the Tweets were all taken from events which were of the same type (such as quick, local disasters), many of the Tweets would share similar linguistic patterns. For example, many of the Tweets might be of the form "Some event happend at some location at some time". Therefore, if the classifier learned to correctly label sentences of that form, it would have an artificially high evaluation score. It was thought that an appropriate mix of subject-oriented posts and random posts would allow for a more accurate and useful evaluation than either a completely random sample of posts or a completely targeted sample.

4 Annotation

A corpus of annotated Twitter postings was made from the 1,684 posts described above. These postings were hand annotated using the Knowtator tool created by Philip Ogren. As much as possible, these Tweets were annotating using the CoNLL-2003 guidelines. Ideally, the Twitter annotation would follow identical guidelines so that difficulties in transfer learning between the CoNLL-2003 trained classifier and the Twitter dataset would be completely due to differences in the text. However, there are several occasions where the Twitter text is not amenable to the guidelines. For example, Twitterers often refer to other Twitterers by the screen name. It was decided that these mentions should be marked as named entities, even though there is some question as to whether or not they would be in the CoNLL guidelines. There were several other ambiguities in annotation guidelines, all of which were resolved by attempting to do what would best match the CoNLL dataset.

The size of the dataset was designed such that it would result in similar evaluative power as the CoNLL-2003 shared task test set. This means that innacuracies in the results of evaluations on the Twitter set will largely be due to annotation inconsistencies and systematic sampling errors instead of statistical uncertainties associated with a small dataset. The Twitter dataset contains 30,289 tokens and 1,743 entity mentions. This significantly lower than the 46,435 tokens and 5,648 mentions of the CoNLL set. However, the error introduced by annotation inconsistencies between the CoNLL dataset and the Twitter dataset probably produces much larger inaccuracies than those due to smaller sampling size.

5 Approach

A Support Vector Machine implementation, YamCha, was used on feature augmented data to classify the entities. This approach has been used in many conference shared-tasks, such as MUC, CoNLL, and ACE, successfully for standard named entity recognition.[16] Standard feature streams, such as noun chunk tags, part of speech information, gazetteer matching, and others were implemented. In addition, experimental streams, such as a measure for morphological regularity, were tested. Additionally, an automatic gazetteer generation algorithm discussed by Toral and Munez and Kazama and Torisawa was used to create gazetteers.[17][8]

5.1 Algorithms

YamCha's Support Vector Machine implementation was used to classify entities. Support Vector Machines classify data by constructing an optimal decision hyperplane through an n-dimensional feature space representing a set of data points. In our task, each word is given value for a set of features. An example feature is part of speech, where 'Verb' might be given the value 0 and 'Proper Noun' given the value 12. The set of all the features is combined and called the vector. The decision hyperplane is constructed so as to maximize the distance between the cluster of one type from the cluster of the other type. The hyperplane is determined by examining the vectors that lie closest to the hyperplane, called the Support Vectors. Support Vector Machines are extensively discussed by Burges and others.[4]

5.2 Feature Streams

Several feature streams were extracted from the text to help the classifier. Many of the feature extractors were written with the help of the open source Natural Language Toolkit.[9] Below is a table containing showing the features which were used.

Feature	Generation Technique
Token	taken from incoming text
Part of Speech	MaxEnt classifier trained on the Penn Treebank dataset [15]
Noun Chunking	MaxEnt classifier trained on the CoNLL 2000 chunking dataset [5]
Word Stem	Porter word stemming algorithm [13]
Word Suffixes	matching from Wiktionary list of suffixes
Type Classification	separated into 9 categories such as 'AllCap', 'NoCap', and 'Numeric'
Dictionary Lookup	checked to see if token appeared in the Unix wordlist include in FreeBSD
Gazetteer Matching	matched to gazetteer lists generated from Wikipedia (see below)
Twitter Bigrams	bigram likelihoods based on 7000 Twitter postings(see below)
Google Bigrams	bigram likelihoods based on truncated GoogleNGram corpus
Google Letter Bigrams	bigram likelihoods based on truncated GoogleNGram corpus

The Google N-gram corpus is a free data set released by Google. The complete data set contains 314,843,401 word bigrams. Due to the size of the dataset (about 24 gigabytes) a condensed version, which was made available by Peter Norvig, was used. It contains the 250,000 most common word bigrams. In addition, it contains bigrams for each combination of letters. These letter bigrams were generated from the entire corpus (approximately 1 trillion tokens).

A feature window of -2...2 was used. This means that the token and all of the features associated with that token were used as well as the preceding two tokens and their features and the following two tokens and their features when determining the Named Entity category for each token. Additionally, the Named Entity tags of the preceding two tokens are also used.

5.3 Gazetteer Generation

Previous research from myself and others has shown the gazetteers are extremely important in creating a competitive named entity recognizer.[12][10] Additionally, due to the nature of discourse on Twitter, more extensive lists will need to be generated in order to cover the plethora of subjects and forms that Twitterers use. Therefore, it is expected that large gazetteers with more forms, but potentially lower precision, will enhance the performance of a named entity recognizer when compared to a smaller set of gazetteers. Since generating new gazetteers is very laborious, an automatic method for creating gazetteers was needed.

Wikipedia is a prime candidate to generate such lists because it is extensive and fairly well structured. Several approaches to utilizing Wikipedia in named entity recognition and gazetteer generation have been proposed. Toral and Munoz used Wordnet and Part of Speech data to process text from Wikipedia pages to attempt to compile lists of entities.[17] Richman and Schone used the Category structure of Wikipedia to extract multilingual data to make a cross-language entity recognizer.[14] A hybrid of these two approaches was used: The category structure was manipulated to generate a lower precision entity list.

Wikpedia contains a category hierarchy to help classify articles. Categories are created for almost every type of article which is present on Wikipedia. The categories are organized into a hierarchy of supercategories and subcategories. For example, "Geography" is a supercategory to "Geography by Country" while "Geography in the United Kingdom" is a subcategory to "Geography by County".

Wikipedia also contains many pages which are lists of things. For example, there is a page titled "Lists of Airports in the United Kingdom". On this page, there is a table which contains a list of all of the airports in the United Kingdom. Almost every entry in the table is a link to the Wikipedia page about the specific airport. Additionally, their are very few links on the page that are not links to airports. By collecting the text from all of the links from list pages, we can get a reasonably accurate list of entities. The links can be further refined using a series of regular expressions to eliminate common links which are known not to be entities or to be entities of a different type (one example of this is that lists of buildings often contains links to the location of the building).

Conveniently, Wikipedia contains categories of lists which contains subcategories of more specific lists of the same time. For example, Wikipedia has a category structure for "Lists of Places" which contains sub-categories all the way down to "Lists of Watermills in the United Kingdom". By performing a tree traversal on this category hierarchy, we can generate a comprehensive set of lists about a certain topic, and then use the method described above to pull the entities out of those lists. This method can be used to generate gazetteers about any topic for which there is a category on Wikipedia, which could be extremely helpful in situations where gazetteers about a specific topic. Alternatively, when the process is applied to a very general category such as "Lists of Places", a very general list of locations can be generated. This creates gazetteers which have a very large coverage over the entire topic, which is a common deficiency of hand made gazetteers.

When used to generate gazetteers for the named entity recognition on Twitter, I was able to create a gazetteer of locations which contains 150,000 entries using the root category "Lists of Places". Additionally, a gazetteer of organizations was created using "Lists of Companies" which contains 40,000 entries.

6 Results

The performance of this system is broken down into three major divisions. First, the performance of the system when using all features is analyzed on both Twitter and CoNLL data. Second, each feature is analyzed to attempt to determine how much each feature contributes to the accuracy of the system for both types of text. Lastly, the performance of a classifier using the same features, but trained on part of the Twitter data instead of the CoNLL data, is analyzed to attempt to quantify the benefits of annotating a new corpora against adapting a classifier from a different domain.

6.1 Twitter and CoNLL Performance

A support vector machine was trained on the CoNLL training using all of the features listed above (parts of speech, noun chunks, word stems, suffixes, token types, gazetteer matches, dictionary matches, Twitter N-gram information, Google N-gram information, and Google letter N-gram information). The classifier was then run over the CoNLL test sets and the Twitter test set. Performance on the CoNLL sets was competitive with systems developed for the 2003 shared task, while performance on the Twitter dataset was very low.

On the CoNLL developers test set (testa) and evaluation test set (testb), the system scored F1 measures of 88.19 and 83.25 percent. This place the system in the middle of the pack for 2003 shared task submissions. Interestingly, the system scored very high F1 measures in classifying locations (less than 3 percentage points below the best performing system). This is likely due in part to the extensive location gazetteer generated using the method described above. Performance in classifying people and miscellaneous categories were further behind the state of art.

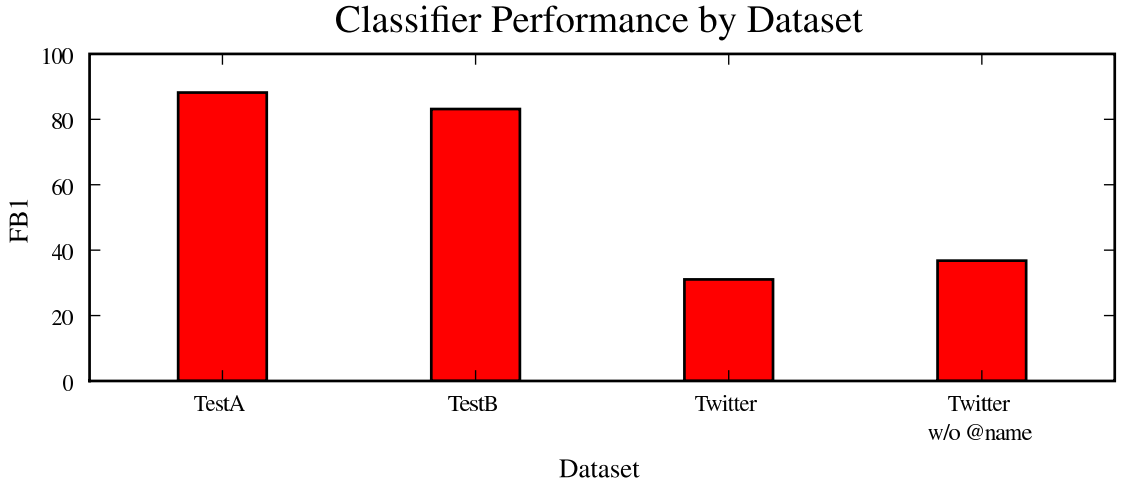

Figure 1: Comparison of classifier performance when using all features on various datasets.

Performance on the Twitter dataset was extremely poor. Overall, the system scored an F1 measure of just 31.05 percent. The system scored significantly better on locations (F1 of 53.65 percent), while it had the most trouble classifying people (F1 of 18.23 percent). There are several reasons why these scores are so low. Firstly, Twitter posts are more polysemous than newswire text. For example, locations often double as channel labels. In the Bozeman gas explosion section of the corpus, users often tagged their posts with '#bozeman' to signify that it related to the gas explosion. Usages of this type were not marked as locations in the annotation. Another example is that users often address their Tweets to other Twitterers by using '@' followed by the username. Usages of this type are very difficult for the classifier to infer, because there is no equivalent usage in newswire texts. If this usage of usernames is not marked as a person, the performance of the classifier increases to an F1 of 32.7 percent on people.

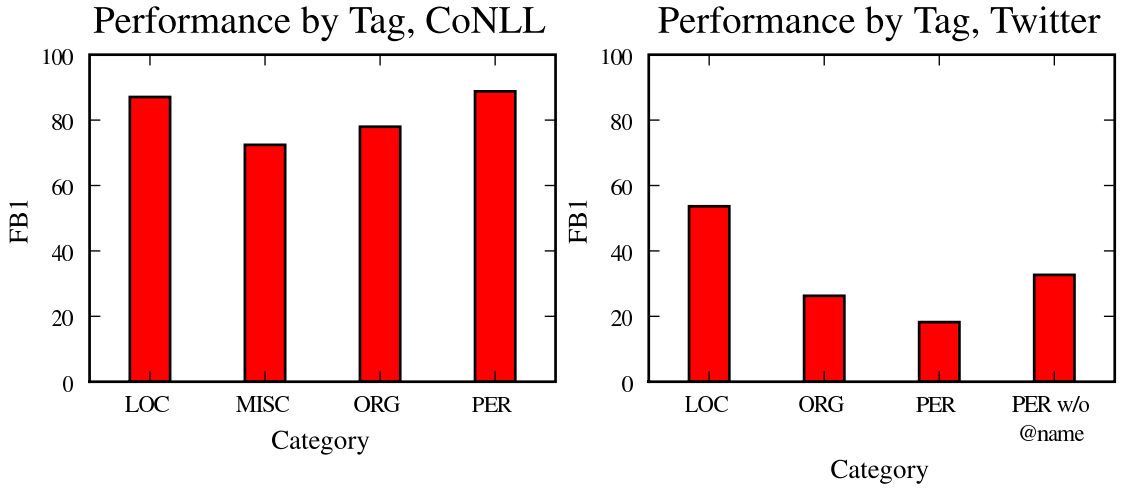

Figure 2: Comparison of classifier performance across entity types.

Even with the inclusion of bigram information from Twitter, the classifier still performs extremely poorly. However, the systems performance could likely be increased easily by using post-processing rules, such as marking all tokens which follow the '@' symbol as persons. This suggests that the most efficient way to adapt a system to a new domain may be by incorporating knowledge in the form of rules, as opposed to modifying the characteristics of the classifier.

6.2 Individual Feature Analysis

Several classifiers were then trained with differing sets of features used, with the intent of clarifying which features contribute to the accuracy of the classifier. The features were separated into two groups: Features which are so inexpensive to calculate that they would be included in nearly any conceivable implementation and features which are more expensive to generate. Part of speech tags, noun chunking information, word stems, and suffixes were included in the first category, while gazetteer matches, dictionary matches, Twitter N-gram, Google N-gram, and Google letter N-gram information were determined to be more expensive to calculate. Classifiers were trained which used only the token information and 1 of the easy-to-calculate features. This had the effect of isolating the contribution of that feature. In order to determine the contribution of the more expensive features, classifiers using the token, all of the easy features, and one of each of the expensive features were trained. By contrasting the contribution of each of the features between the datasets, it is possible to see which types of features should be pursued in adapting the system to less structured domains.

Surprisingly, token type information (classifications such as "StartsWithCap") is the most effective simple feature in both the Twitter data and in the CoNLL data. It is also the least expensive feature to calculate. It can be seen that Part of Speech tags contribute greatly. Interestingly, however, the contribution of suffix information is nearly as large as Part of Speech information in the Twitter corpus. In the CoNLL corpus, it is not nearly as helpful. This may be partially due to suffix extraction being equally accurate regardless of text type, whereas the accuracy of the part of speech tags is likely lower on microblog text. However, stemming information has the same characteristic, but it was not more helpful in the Twitter classification than in the CoNLL classification.

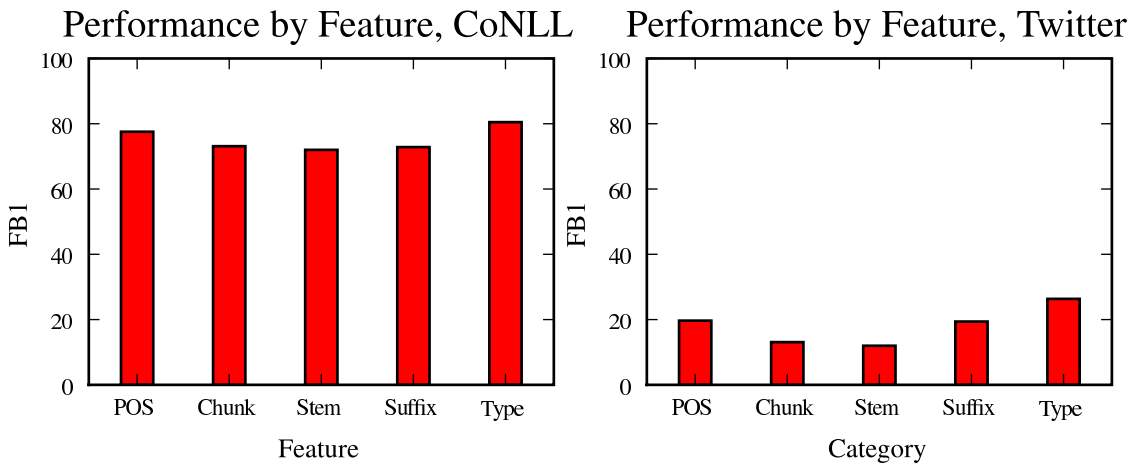

Figure 3: Comparison of classifier performance when only certain features used.

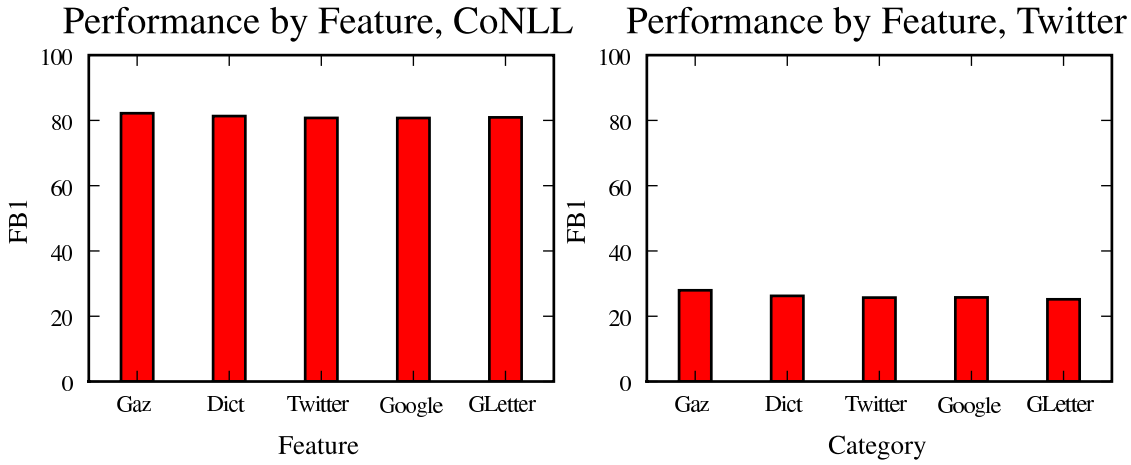

Figure 4: Comparison of classifier performance when only certain complex feature and all simple features used. Gazetteer matching (Gaz), Dictionary Matching (Dict), Twitter Bigram Information (Twitter), Google Word Bigram Information (Google), and Google Letter Bigram information (GLetter) compared.

As expected, gazetteer information was the most helpful complex feature in both the Twitter task and the CoNLL task. It is interesting that dictionary matching was also fairly effective in both cases, despite dictionaries being cheap (in terms of work hours) to create when compared with gazetteer lists. Also, neither the Google bigrams nor the Twitter bigrams were as helpful creating an adaptive system as might be expected.

It should be noted that analysis of this technique has a major flaw. It does not take into consideration the interplay between features. For example, if one feature is particularly good at disambiguating between two very similar feature vectors in a specific case, its contribution would be undervalued by this method. A method that would more appropriately address this defect would be training classifiers which contain all of the features except 1 in order to isolate that feature's contribution. Due to time constraints, analysis of this type was not performed.

6.3 New Corpora vs. Adaptation

Since the CoNLL trained classifier performed so poorly on the Twitter data, it became conceivable that a classifier trained on *much* less data might be able to perform better. In order to investigate this hypothesis, the Twitter dataset was split into a training set containing 24,000 tokens and a hold-out set which contained the remaining 6,283 tokens. A classifier was trained on the training set and evaluated on the hold-out set. The results are much less significant since the test set was so small.

Type	Precision	Recall	F1
TOTAL	74.61	49.48	59.50
LOC	50.00	28.07	35.96
ORG	57.14	24.49	34.29
PER	87.39	76.47	81.57

The annotation was created in less time than it took to develop the named entity system. This suggests that the most efficient way for an organization to develop a named entity recognition system for use in a new domain might be to create their own dataset, even if that dataset is very small. However, a very good transfer learning algorithm could conceivably be used in many domains at relatively low additional cost, while a new corpus would need to be created for each new domain using the new dataset approach. Additionally, the performance of the Twitter-trained method is inflated when compared with the CoNLL-trained because there is absolute annotation consistency between the Twitter training and hold-out sets, while there is not between the Twitter set and the CoNLL set.

7 Conclusions and Future Work

In this paper, the performance of a named entity classifier was evaluated on CoNLL 2003 shared task data and Twitter posts. The classifier was able to perform very well on the CoNLL data, but performed very poorly on the Twitter data. This suggests that unstructured microblogging data and newswire text are so different that it may be difficult to develop a classifier which can transfer learning from one domain to the other.

The ability to tag locations transfered from the newswire text to the microblogging text better than the ability to tag organizations or people. In both datasets, token type information and gazetteer match information improved the accuracy of the classifier greatly when included. Part of Speech information was not as helpful in the Twitter corpus as expected. Additionally, N-gram features based on both Google and Twitter were not particularly helpful.

There are several promising approaches to improve the performance of the system on Twitter posts. Post processing rules which take advantage of conventions of Twitter, such as hashtags signifying a channel of conversation (and therefore not a location) could provide some improvement. In addition, implementing more advanced transfer learning methods, such as WordNet hierarchy manipulators, would likely improve performance further.[1] However, to create a highly accurate named entity, very large improvements would need to be made. Given the encouraging performance of the system when trained on a small Twitter corpus, it may be best to invest effort in creating a new dataset for microblog postings.

8 About This Document

This document was created using Latex in TeXnicCenter. The bibliography was created using BibTex. The template for this document was originally created by Nicolas Nicolov. The charts were created using MatPlotLib.

Bibliography

[1] Andrew Arnold, Ramesh Nallapati, and William W. Cohen. Exploiting feature hierarchy for transfer learning in named entity recognition. 2008.

[2] Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains: A case study. In *RANLP*, 2005.

[3] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[4] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[5] Orlando Alberto Carvajal. A hybrid symbolic-numeric method for multiple integration based on tensor-product series approximations. In *In: Proceedings of CoNLL-2000 and LLL-2000*, 2004.

[6] Nancy Chinchor. Muc-7 named entity task. In *Proceedings of MUC-7, 1997*, 1997.

[7] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *Proceedings of LREC 2004*, pages 837–840, 2004.

[8] Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, 2007.

[9] Edward Loper and Steven Bird. Nltk: The natural language toolkit, May 2002.

[10] Richard Morgan, Roberto Garigliano, Paul Callaghan, Sanjay Poria, Mark Smith, Agnieszka Urbanowicz, Russell Collingham, Marco Costantino, Chris Cooper, and LOLITA Group. University of durham: Description of the lolita system as used in muc-6. In *In Proc of the MUC-6, NIST, Morgan-Kaufmann Publishers*. Morgan Kaufmann Publishers, 1995.

[11] Lisa Ferro Nancy Chinchor, Erica Brown and Patty Robinson. 1999 named entity recognition task definition, 1999.

[12] David Palmer, , David D. Palmer, and David S. Day. A statistical profile of the named entity task. In *Proc. ACL Conference for Applied Natural Language Processing*, pages 190–193, 1997.

[13] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[14] Alexander E. Richman and Patrick Schone. Mining wiki resources for multilingual named entity recognition. In *Proceedings of ACL-08: HLT*, pages 1–9, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[15] Ann Taylor, Mitchell Marcus, and Beatrice Santorini. The penn treebank: An overview, 2003.

[16] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[17] A. Toral and R. Munoz. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. *EACL 2006*, 2006.