Using an Automatically Generated Dictionary and a Classifier to Identify a Person's Profession in Tweets

Abe Hall and Fernando Gomez

School of Electrical Engineering and Computer Science, University of Central Florida
abe.hall@knights.ucf.edu, gomez@eecs.ucf.edu

Abstract

Algorithms for classifying pre-tagged person entities in tweets into one of 8 profession categories are presented. A classifier using a semi-supervised learning algorithm that takes into consideration the local context surrounding the entity in the tweet, hash tag information, and topic signature scores is described. A method that uses data from the Web to dynamically create a reference file called a person dictionary, which contains person/profession relationships, is described, as is an algorithm to use the dictionary to assign a person into one of the 8 profession categories. Results show that classifications made with the automated person dictionary compare favorably to classifications made using a manually compiled dictionary. Results also show that classifications made using either the dictionary or the classifier are moderately successful and that a hybrid method using both offers significant improvement.

Introduction

Twitter is becoming increasingly popular and potentially offers a wealth of information. For this reason, an increasing amount of work is being done to produce good natural language processing tools for use with Twitter and other social media sites. Because social media text generally contains grammatical errors, misspellings, non-standard abbreviations, and meaningless capitalization (Ritter, Clark, and Etzioni 2011), dealing with this type of informal text is more complicated than dealing with formal text. Twitter also presents an added challenge since tweets are limited to a maximum of 140 characters.

Named Entity Recognition (NER) is one area where widely used tools do not translate well to tweets, but recent work in this area has shown an increased ability to accurately tag entities in tweets (Li et al. 2012). However, in this paper, our primary concern is not the initial entity recognition that classifies an entity into a coarse grained category such as Person, Location, or Organization. Rather, we focus on assigning an already tagged Person to one of eight subcategories representing classes of professions. This information can then be used not only in Question/Answering systems, but also as a method of filtering twitter users based on who they tweet about.

To do this, we employed two strategies. The first was a semi-supervised learning method adapted from research done on fine grained person classification (Fleischman and Hovy 2002). The second strategy was an unsupervised method that used the Web to dynamically build a resource containing person/profession relationships, referred to as a person dictionary. This dictionary is then used to make the profession classification. We will describe our algorithm for automatically acquiring data to build our person dictionary as well as our method for using the dictionary to assign a profession category. Lastly, we will explain how we were able to use both approaches together to achieve a high level of classification accuracy.

Background

There are two main approaches to assigning a fine grained classification to an entity. The first is to train the recognition tool in more than the basic entity categories. Some NER tools can accurately assign entities in as many as 100 categories (Nadeau, 2007). However, because of the generally low quality of tweet text, traditional NER tools trained on formal corpora such as newspaper articles do not perform as well on tweets (Locke and Martin 2009). Because of this, we decided not to pursue this method.

The second method is to take data that has been tagged at a coarse grained level and then to sub-classify it further (Fleischman and Hovy 2002). This method relies upon an accurate initial classification of an entity into a coarse grained classifier. Recent work to do this for tweets has been promising. For example, a method that combines a K-nearest neighbors classifier with a Conditional Random Fields model has shown increased accuracy over traditional NER methods (Liu, Zhang, and Wei 2011). Other work has shown that using data from Wikipedia and the Web N-Gram corpus can improve a classifier's ability to identify named entities by providing additional context not available from the tweet itself (Li et al. 2012).

Data Set Generation

Our corpus is a set of 7 million tweets collected via the twitter API sample interface beginning on March 14, 2012 and ending on April 10, 2012. Once collected, duplicate tweets, non-English tweets, and retweets were eliminated from the corpus. To improve the quality of tweet text, we compiled an abbreviations dictionary and a misspellings dictionary. We then used them to replace abbreviations and misspellings in the corpus. The tweets were then stemmed, tokenized, and entity tagged using the Stanford NER (Finkel, Grenager, and Manning 2005). Our final corpus contained 1.2 million tweets with a tagged person entity.

From the corpus, 100 tweets were manually annotated for each of the eight profession categories to be used as seed data. Next, we compiled a person dictionary of over ten thousand entities. The classifier was trained on the seed data and then run on the remaining corpus. Any person classified with a confidence greater than a given threshold that also matched an entry in the person dictionary was added to the training set; but if it did not match a dictionary entry, it was saved and manually verified before being added to the training data and the person dictionary.

For testing, two test sets of tweets were generated. The first set was a manually compiled set of tweets that was removed from the corpus prior to the bootstrapping method that generated the training set. This set ended up containing about 1300 entities and is referred to as the Hold Out Set. The purpose of this set was to test the generalizability of the classifier. The second set consisted of 1000 tweets removed at random from the training data and was used for the initial validation of the classifier. This set is referred to as the Validation Set. After these tweets were removed, the training set contained about 12000 training instances.

Method

Our method takes as input a tweet that has already been entity tagged for Persons and then assigns that person into one of eight categories: Athlete, Businessman, Clergy, Doctor/Scientists, Entertainers, Journalists, Authors, and Politicians/Government Officials. These categories were selected because of their relative frequency in the corpus.

We first used the training data obtained from the bootstrapping method to train a classifier. We then ran the classifier over the test set and any classification with a confidence level greater than a configurable threshold was saved in a table. Through experimentation, we found the ideal value for this threshold to be 82.5%. Next, we looked for any other instances of that person in the set and assigned the same profession to the other instances. We did this based on the assumption that subjects in Twitter trend, so tweets collected in the same time frame that mention the same name most likely are referencing the same person.

Next we ran our person dictionary algorithm. We used the Yahoo! BOSS web search API to query the name of the person, extract the potential professions of the person from the query results, and then save the information in a person dictionary. If there was only one possible profession found, the person was assigned that classification. If there were multiple dictionary entries, then we used a disambiguation metric to score the possible professions. The person was then classified as whichever profession scored the highest.

In the hybrid method, any person with only one entry in the dictionary was assigned that profession as the final classification. Then all other persons with a classification confidence greater than a threshold, experimentally found to be 60%, were assigned the classifier result and any person with a classification confidence lower than that was assigned using the dictionary disambiguation algorithm. If a person had no entries in the dictionary, then the highest scoring classifier classification was used. These steps are described in more detail later in the paper.

Classifier Features

The first two types of features used by our classifier were word frequency features and topic signature score signatures (Fleischman and Hovy 2002). These will only be discussed briefly since we did not change them from the referenced work. We then added hash tag features and used pattern matching to improve the word frequency features.

Word Frequency Features

Word frequency features measure the frequency of words directly before and after each person instance. For each person in a tweet, the three unigrams directly preceding and following the entity were used. The preceding and following bigrams were also used. Lastly, the preceding and following trigrams were used. A frequency table tracked how often each of the word features appeared in relation to each profession. These frequency counts were then used as features for each person, with the count of each profession type being used for each of the ten n-gram types. This produced a total of 80 features per person in each tweet (Fleischman and Hovy 2002).

Topic Signature Scores

The classifier also uses topic signature score features (Lin and Hovy 2000). Using the word frequency counts, each word has a λ-score calculated for it (Dunning 1993) and is stored in a table. This score quantifies the likelihood that the word is an indicator of the profession in question. Once this table of λ-scores was compiled, the topic signature score was calculated for each person/profession combination using the equation:

$$\sum_{N} \frac{\lambda - \text{score of } word_{n,type}}{(distance\ from\ entity)^2}$$

where N is the number of words in the tweet. This raised the total feature count to 88. (Fleischman and Hovy 2002).

Hash Tag Features

Hash tags were used in two ways. The first was that hash tag frequencies were tracked. Each hash tag was tracked in a frequency table similar to the one tracking word features. However, for any given tweet, a varying number of hash tags can be given. This prevented us from creating a separate frequency feature for each unique hash tag in a tweet as doing so would have resulted in a variable number of features for each person instance. Instead the sum of the frequency counts for all the hash tags was used as a feature. This resulted in a total of 96 classifier features.

The second change implemented was in the topic signature score calculation. Originally, a word's λ-score was weighted based on the word's distance from the entity. We altered the calculation so that each λ-score calculated for a hash tag was fully weighted in the calculation.

Pattern Matching

We also used pattern matching (Hearst 1992) to improve the word frequency features. One issue with the word frequency features is that it uses the n-grams exactly as they appear in the tweet text. For example, in the tweet below, the score "88-85" would originally have been saved as "88-85" in the frequency table.

> Lakers beat Hornets 88-85 on Bryant's late 3 (Yahoo! Sports) : Kobe Bryant hit a go-ahead 3-pointer with 20 secon...

But using pattern matching, "88-85" would be replaced with a <SCORE> tag. This allowed all scores to be tracked with one entry, improving the word frequency features. We also modified the topic signature score calculation for the patterns to use the full weight of the λ-score in the calculation. We tested several patterns, but ultimately only two affected the results significantly. Those patterns were scores and ratings. Ratings were when users reviewed something and gave it a numerical rating.

Creating and Using the Person Dictionary

Manually compiling the person dictionary used for the training set generation was tedious and time consuming. So we set out to automatically build the dictionary using the Web. First, we created a reference file wherein the eight profession categories were mapped to specific occupations. For example, in this file "basketball player" and "football player" would both be linked to the Athlete category.

For our automatically generated dictionary, we started with an empty file. Then, for each person in the test sets, we checked the person dictionary to see if an entry already existed. If there were no entry, a query for the person's name was run using the Yahoo! BOSS search API. Any web result containing the pattern "<Person's name> is", "<Person's name> was", or "<Person's name> became" was stored for processing. Additionally, any web result from Wikipedia was stored for processing.

To process the web results, we looked at only the sentence that contained the pattern. We POS tagged the sentence and then extracted all the nouns following the pattern. For any noun that matched one of the occupations in the occupation-profession relationship file, an entry was added to the person dictionary for that person.

In order to make this algorithm more robust, we then created patterns that ignored middle names, birth and death dates, and suffixes. Also, we created a file that contained common nicknames such as "Lucy" in place of "Lucille." This allowed us to parse web results for pattern matches using both nicknames and full names.

Disambiguation Metric

Ambiguity was introduced into the person dictionary in one of two ways. The first was when multiple people share a name and are known for different professions. For example, a search for "Michael Lewis" would return a "Michael Lewis" who is an athlete, another who is a journalist, and another who is an author. The second way was when a person is known for multiple professions. For example, a search on "Mitt Romney" would return that he is both a businessman and a politician. In these situations, we considered the correct classification to be the one for which the person is best known.

To deal with the ambiguity in the dictionary, we calculated the word similarity score (Resnik 1999) between each word in the tweet against each profession category, omitting all proper nouns:

$$wsr(w_t, w_c) = \max_{c_t, c_c}[srm(c_t, c_c)]$$

where *srm* is the Resnik similarity measure and c_t, c_c represents a sense of the target word (w_t) and the category word (w_c) respectively.

We then normalized each score by determining what the maximum possible similarity score for each category would be, and then multiplying the ratio of the similarity score of the word by ten. The maximum possible score was calculated as the word similarity between the category and itself. For example, the word similarity score between "athlete" and "athlete" is 6.789.

$$wsr_{norm}(w_t, w_c) = \frac{wsr(w_t, w_c)}{wsr(w_c, w_c)} * 10$$

The final score was calculated as the sum of all the normalized word similarity scores in the original tweet.

$$sim\_score_c = \sum_i wsr_{norm}(w_i, w_c)$$

The category that ended up with the highest score was then assigned as the profession category for the person entity.

Results

The results of the classifier on the validation and hold out sets are shown in Figure 1.

	Validation	Hold Out
Word Frequency (WF), Topic Signature (TS)	71.7%	56.3%
WF, TS, Pattern Matching (PM)	78.2%	67.1%
WF, TS, PM, Hash Tag Features	80.1%	71.1%

Figure 1: Accuracy broken down by Feature Combinations

The results show the additional features were beneficial to the classifier. We think that the score tag enabled the classifier to better distinguish between Entertainers and Athletes and to a lesser extent to identify Politicians (because vote totals match the score pattern). Also, we noticed a significant number of book reviews in the testing sets which is why the ratings pattern helped as well. This helped the classifier to choose between Author and Journalist. The hash tag features also helped minimally, but we think the majority of that benefit was derived from the topic signature score calculation change.

The results of the dictionary classification method are shown in Figure 2. For persons with only one possible profession in the dictionary, the results show high accuracy but also indicate that the algorithm is not always finding the correct option from the web. For persons with multiple profession entries in the dictionary, the results show that the accuracy suffers as the number of professions increases, but the accuracy is still far better than a random selection. The table also shows the results when run with the manually compiled dictionary that was used for producing the classifier training set. The automated version of the dictionary compares favorably to the more extensive manual dictionary while also indicating that there is still room for improvement in the algorithm.

	Validation	**Hold Out**
With 1 Profession Entry	96.7%	97.6%
With 2 Profession Entries	79.9%	73.8%
With 3 Profession Entries	72.2%	59.5%
Total Accuracy	83.7%	76.2%
Total Accuracy w/ Manual Dictionary	89.9%	85.0%

Figure 2: Person Dictionary classification method accuracy

In Figure 3, we show the results of the hybrid classifier. The results show that while both the classifier and dynamic person dictionary method are modestly successful by themselves, the hybrid method produces far better results.

	Validation	**Hold Out**
Classifier	80.1%	71.1%
Person Dictionary	83.7%	76.2%
Hybrid	92.3%	88.0%

Figure 3: Accuracy of the hybrid classification method

Conclusions

We have presented a method that uses a classifier and a dynamically generated dictionary to assign one of eight profession categories to a person in tweets. We have shown hash tag features and pattern matching can improve the accuracy of the classifier. We have also shown that the Web can be used to produce a person dictionary that compares favorably to more extensive, manually compiled dictionaries although further work is still needed to improve the algorithm that generates this dictionary. These dictionaries can then be used to make accurate profession classifications. We believe the work with the person dictionaries is particularly useful as they could eventually be used as resources for other projects. Lastly, we have shown that a hybrid classifier that uses the strengths of both methods produces substantially more accurate results.

References

Cucerzan, S. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 708-716. Prague, Czech Republic.

Dunning, T.E. 1993. Accurate methods for statistics of surprise and coincidence. *Computational Linguistics* 19(1): 61-74.

Finkel, J.R.; Grenager, T.; and Manning, C. 2005. Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of ACL 2005, 363-370. Ann Arbor, Michigan.

Fleischman, M. 2001. Automated Subcategorization of Named Entities. In Proceedings of the ACL Student Workshop, 25-30. Toulouse, France.

Fleischman, M. and Hovy, E. 2002. Fine Grained Classification of Named Entities. In Proceedings of COLING-02, 1-7. Taipei, Taiwan.

Hearst, M. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In Proceedings of the Fourteenth International Conference on Computation Linguistics, 539-545. Nantes, France.

Li, C.; Weng, J.; He, Q.; Yao, Y.; Datta, A.; Sun, A.; and Lee, B-S. 2012. TwiNER: Named Entity Recognition in Targeted Twitter Stream. In Proceedings of SIGIR 2012. Portland, Oregon.: ACM 978-1-4503-1472

Lin, C-Y. and Hovy, E. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In Proceedings of COLING-00, 495-501. Saarbruken, Germany.

Liu, X.; Zhang, S.; and Wei, F. 2011. Recognizing Named Entities in Tweets. In Proceedings of the Association for Computational Linguistics: Human Language Technologies, 359-367. Portland, Oregon.

Locke, B. and Martin, J. 2009. Named Entity Recognition: Adapting to Microblogging. Senior Thesis, Department of Computer Science, University of Colorado, Boulder, CO.

Nadeau, D. 2007. Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision. Ph. D. diss., Ottawa-Carleton Institute for Computer Science School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada.

Ritter, A.; Clark, S.; and Etzioni, O. 2011. Named Entity Recognition in Tweets: An Experimental Study. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 1524-1534. Association for Computational Linguistics.