

Automated Subcategorization of Named Entities

Michael Fleischman

USC Information Science Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
U.S.A.
fleisch@ISI.edu

Abstract

There has been much interest in the recent past concerning the possibilities for automated categorization of named entities. The research presented here describes a method for the subcategorization of location names. Subcategorization of locations is not a trivial task even for human subjects, who perform at accuracy levels of less than 58%. After experimenting with both Bayesian classifiers and decision tree learning algorithms, we have designed a system that achieves accuracy levels greater than 80%.

1. Introduction

There has been much interest in the recent past concerning automated categorization of text elements. We focus on the classification of specific elements within text, such as named entities. Recent advances have brought some systems (such as BBN's IdentiFinder (Bikel, 1999)) to within 90% accuracy when classifying named entities into broad categories, such as person, organization, and location. While the accurate classification of such general named entities is useful in many areas of natural language research, more fine-grained categorizations would be of particular value in those areas concerned with semantic content, such as question-answering, information retrieval, and the automated construction of ontologies. The research presented here is an initial effort to further advance the automated classification of named entities.

We have chosen to focus on the subcategorization of location names as an initial effort in our research. Location names are a logical choice, as the availability of precompiled lists of names for many subcategories simplifies the task of constructing a

training set. Further, location names can be subcategorized into well-defined and non-overlapping categories, unlike person names that are often harder to characterize. However, the subcategorization of location names is not a trivial task even for human subjects. Below are examples of sentences in which the names of locations have been encrypted using a simple substitution cipher. The names are of only three subtypes: region, city, and territory, yet prove remarkably difficult to classify based upon the context of the sentence.

1 "This destructive virus is spreading world-wide very rapidly," read one January posting on a *Vsaogptmos* _____ board.

2 The pulp and paper operations were moved to *Dpiyj Vstpaoms* _____ in 1981 .

3 The company , which is based in *Dpiyj Dsm Gtsmdodvp* _____, *Vsaog.* _____ said an antibody prevented development of paralysis in about 70% of the treated rats , and delayed and reduced the degree of paralysis in the other cases .

4 "The situation has been strained ever since *Yplup* _____ began waging war in *Dpiyj RsdY Sdos* _____."

1 *California-territory*

3b *Calif.-territory*

2 *South Carolina-territory*

4a *Tokyo-city*

3a *South San Francisco-region*

4b *South East Asia*

In this work we investigate the power of two feature based learning algorithms, Bayesian classifiers and decision trees, to perform the automated subcategorization of location names.

2. Methods

Experiment 1: Bayesian Classifier

A training data set containing approximately 16000 sentences (24314 location instances) was compiled from a TREC9 database consisting of articles from the Financial Times of London, Mercury Sun Times, and Wall Street Journal. In each data set the sentences are stripped of their HTML tags, word tokenized, and tagged by BBN's

classification program, *IdentiFinder*, which classifies proper names in a sentence as a location, a person, or an organization. The location instances are then put through a partially automated classification system that forces a further designation of locations into one of eight categories: Country, City, Street, Territory, Region, Water, Mountain, and Artifact (e.g., man-made locations, such as airports and universities). Each instance is then paired with a set of 11 features, each feature corresponding to the lexical item found in one of 11 positions surrounding the target instance. The positions include the three individual words before the occurrence of the instance, the three individual words after the instance, the 2 word bigram immediately before and after the instance, the three word trigram immediately before and after the instance, and, if the instance is made up of more than one word, the most frequent word in the instance itself.

The Bayesian classifier works on the principle that finding the most likely classification given a feature set can be reduced to finding the maximum, for each classification, of the probability of that classification multiplied by the probability of a feature set given that classification:

$$\operatorname{argmax}_{\text{class}} P(\text{class} \mid \text{feature set}) = \operatorname{argmax}_{\text{class}} P(\text{class}) * P(\text{feature set} \mid \text{class})$$

The $P(\text{class})$ for each class is computed by simply dividing the count of each instance type by the total count of instances. The $P(\text{feature set} \mid \text{class})$ is derived for a specific class by multiplying together for all i , $P(\text{feature}_i \mid \text{class})$, where feature_i corresponds to the word found in one of the 11 positions surrounding the target instance described above. The $P(\text{feature}_i \mid \text{class})$ is in turn computed by dividing the number of times the feature has appeared with that class by the total count of words that have appeared as a feature for that class; i.e., the number of times that the word seen in that feature position coincided with an instance of that class, divided by the total number of words seen in that feature position for that class.¹ In order to allow for the possibility that a word has not yet appeared as a specific feature with a specific class, the probabilities are smoothed using Witten-Bell smoothing (Jurafsky and Martin, 2000).

¹ The probability for the feature positions internal to the name itself is the maximum probability word within the name.

The resulting Bayesian classifier is then used to estimate the classifications of a group of instances in a held out test set. The test set contains approximately 700 sentences (1100 instances) composed from the same set of newspapers with which the classifier was trained and with the same approximate distribution of instance types.

Initial results with this classifier revealed that, due to differing contexts, instances of the same name within a single document would be classified into different subcategories. Often, an instance co-occurs with features that do not determine its true subcategorization. In such cases, the probability with which a particular classification is determined will be relatively low. Therefore, we augmented the classifier with another program, *MemRun*, which standardizes the subcategorization of instances based on their most frequent classification. *MemRun* is based upon the hypothesis that by looking at all of the classifications that a location has received throughout the test set, an “average” sub-categorization can be computed that offers a better guess than a low probability individual classification.

MemRun operates in two rounds. In the first round, each instance in the test set is evaluated by the Bayesian classifier and receives a classification hypothesis. This hypothesis is then entered into a temporary database that records all of the different hypotheses that each instance has received, along with the number of times that each hypothesis has been encountered. When all of the instances in the test set have been examined, the round is complete.

| Instance | Classification | Occurrence |
|------------|----------------|------------|
| New Jersey | City | 4 |
| | Territory | 1 |
| U.K. | Country | 7 |
| | City | 2 |

Figure 1. *MemRun* database for Bayesian classifier

In the second round, the test set is again examined. This time, however, each instance is looked up in the temporary database and the most common classification, i.e., the classification with the highest occurrence value, is returned.

Experiment 2: Decision Tree

The next experiment replaces the Bayesian Classifier by the decision tree learning algorithm, *C4.5* (Quinlan, 1993). In training the decision tree, a series of training sets is used, each with differing numbers of sentences, in order to examine the effect of training size on overall performance. Four

training data sets are generated from a TREC9 database consisting of articles from the Financial Times of London, the Wall Street Journal, and the San Jose Mercury Sun Times. The data sets contain, respectively, 1000, 3000, 5000, 7000, and 16000 sentences. In each data set the sentences are prepared in the same manner as described above.

Each subcategorized location instance is then paired with a set of features that describe that instance. Each feature within that feature set represents how often the word in one of the positions surrounding the target instance occurs with a specific sub-categorization in the training set. For example, in example sentence 1 in the introduction, the word “a” occurs in the position immediately before the location instance. The feature set describing this instance would thus include eight different features; each denoting the frequency with which “a” occurred in the position immediately preceding an instance of a Country, a City, a Region, etc. The feature set includes these eight different frequencies for 11 word positions (totaling 88 features per instance). The positions used are as described in Experiment 2 with the internal position being calculated by summing the frequencies of the words within a multi-word instance (e.g. “East,” + “New” + “Brunswick”, in “East New Brunswick”).

All feature set-classification pairs are then run through the decision tree program C4.5. The results of the C4.5 program are a set of if-then rules that apply to the features of an instance and generate a classification hypothesis of said instance, along with a particular degree of confidence. A test set containing approximately 700 novel sentences (1096 location instances) is generated as above from the same TREC9 database of Financial Times of London articles. The test set is then run through the program, MemRun, which was modified for use with C4.5.

In this experiment MemRun was augmented to take advantage of the confidence levels with which the if-then rules generated by C4.5 make their classification. Again MemRun operates in two rounds. In the first round, each instance of the test set is evaluated using the rule set and a classification hypothesis is generated. If the confidence level of this hypothesis is above a certain threshold (THRESH 1), then the hypothesis is entered into the temporary database (see below) along with the degree of confidence of that hypothesis, and the number of times that hypothesis has been received.

| Instance | Class | Confidence | Occur |
|------------|-----------|------------|-------|
| New Jersey | City | %97.5 | 4 |
| | Territory | %83.4 | 1 |
| U.K. | Country | %92.4 | 7 |
| | City | %72.1 | 2 |

Figure 2 MemRun database for Decision Tree classifier

At each subsequent occurrence of that same instance, the database is updated in the following way. If the same classification hypothesis is received with the same confidence level as before, the increment is advanced and the program moves on. If the same classification is received based on a different confidence level, the increment is advanced and the degree of confidence is averaged between the old and new levels (taking into account the increment of the old confidence level). If a different classification is found, a new entry in the database is created with the new classification, the new confidence level, and the number of times said classification was found. When all of the instances in the data set are examined, the round is complete.

In MemRun’s second round, the data set is reexamined, now with the initial classification hypothesis produced by C4.5. If the probability of a hypothesis is below a second threshold (THRESH 2), then the hypothesis is ignored and the database is consulted. In this experiment, the entries in the database are compared and the most frequent entry (i.e., the max classification based on confidence level multiplied by the increment) is returned. When all instances have been again examined, the round is complete.

Experiment 3: Human Competence

In order to judge human competence on relatively identical tasks, three subjects were given test sets consisting of the first 75 instances from the test set used in Experiment 2. The sentences are identical to those given to the computer but with the letters of the location instances scrambled in a consistent substitution.

3. Results

Experiment 1: Bayesian Classifier

The results of running the Bayesian classifier both with and without the MemRun algorithm are presented in figure 3. MemRun clearly has the greatest effect on Country and Territory subcategories, increasing them both by nearly 15% and 20% respectively. This is not surprising, as these categories often contain a great many instance repetitions, and thus benefit greatly from the averaging of classification hypotheses. While the

use of MemRun does improve most subcategories, it is not without error. Because of its reliance on data from other situations that may not represent the most accurate guesses, MemRun has the capacity to misclassify instances that had initially been correctly classified. This weakness is avoided, however, in experiment 2 by the use of the two thresholds which limit the classifications which can be considered in the averages, and further, limit the instances that rely on those averages.

| Experiment 1 (Bayesian Classifier) | | | | |
|------------------------------------|---------------------|-------------|---------------------|-------------|
| Class | After Bayes | | After MemRun | |
| | # Correct / # Total | % Correct | # Correct / # Total | % Correct |
| City | 130 / 375 | 34.6 | 150 / 375 | 40.0 |
| Country | 298 / 482 | 61.8 | 359 / 482 | 74.5 |
| Street | 5 / 10 | 50.0 | 5 / 10 | 50.0 |
| Region | 20 / 65 | 30.8 | 33 / 65 | 50.8 |
| Water | 3 / 8 | 37.5 | 1 / 8 | 12.5 |
| Artifacts | 1 / 8 | 12.5 | 1 / 8 | 12.5 |
| Territory | 98 / 149 | 65.8 | 128 / 149 | 85.9 |
| Mount | 0 / 3 | 00.0 | 0 / 3 | 0.0 |
| Total | 555 / 1100 | 50.5 | 677 / 1100 | 61.5 |

Figure 3. Overall results for Bayesian Classifier

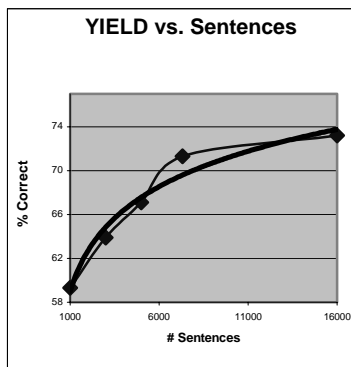


Figure 4. Effects of size of training set on total yield

Experiment 2: Decision Tree

The results of incrementally increasing the size of the training set are illustrated in figure 4. The bold line represents the log of the series and demonstrates the beginning of a plateau effect. It can be hypothesized from this trend that further increases to the training set will provide only marginally improved results.

Figure 5 represents the results of C4.5 and MemRun on the largest data set (16000 sentences, 24314 instances) from Experiment 2. The confusion matrices for the results of MemRun are in 5a. It is again clear that MemRun has a great effect.

Although the results of this experiment are significantly higher than those found using the Bayesian classifier, it is interesting that even after MemRun, the accuracy of C4.5 for the Territory category is significantly lower. This is not easily explained from the present data, but suggests that in fact each system may have advantages that the other does not possess. A possible line of future research then is examining hybrid classifiers that incorporate both these advantages.

| Experiment 2 (Decision Tree) | | | | |
|------------------------------|---------------------|-------------|---------------------|-------------|
| Class | After C4.5 | | After MemRun | |
| | # Correct / # Total | % Correct | # Correct / # Total | % Correct |
| City | 260 / 373 | 69.7 | 309 / 373 | 82.8 |
| Country | 411 / 482 | 85.3 | 440 / 482 | 91.2 |
| Street | 6 / 10 | 60.0 | 6 / 10 | 60.0 |
| Region | 44 / 65 | 67.7 | 44 / 65 | 67.7 |
| Water | 6 / 7 | 85.7 | 6 / 7 | 85.7 |
| Artifacts | 4 / 8 | 50.0 | 4 / 8 | 50.0 |
| Territory | 71 / 148 | 48.0 | 79 / 148 | 53.4 |
| Mount | 0 / 3 | 00.0 | 0 / 3 | 00.0 |
| Total | 802 / 1096 | 73.2 | 888 / 1096 | 81.0 |

Figure 5. Overall results for Decision Tree Classifier

| Actual | CITY | CN | ST | REG | WAT | ART | TER | MOUNT |
|---------|------|-----|----|-----|-----|-----|-----|-------|
| CITY | 309 | 51 | 0 | 3 | 0 | 0 | 10 | 0 |
| COUNTRY | 29 | 440 | 0 | 6 | 0 | 0 | 7 | 0 |
| STREET | 3 | 1 | 6 | 0 | 0 | 0 | 0 | 0 |
| REGION | 12 | 7 | 0 | 44 | 0 | 1 | 1 | 0 |
| WATER | 0 | 1 | 0 | 0 | 6 | 0 | 0 | 0 |
| ARTS | 0 | 3 | 0 | 0 | 0 | 4 | 1 | 0 |
| TER | 32 | 33 | 0 | 4 | 0 | 0 | 79 | 0 |
| MOUNT | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 5a. Confusion Matrix After MemRun

The results of MemRun reported in figure 5 are with THRESH 1 at 77 and THRESH 2 at 98. Thus, in this run, only those classifications with confidence levels above 77% were entered into the database, and only those classifications with confidence levels below 98% are checked by the database. The surface chart for threshold vs. yield shown in figure 6 represents the effect of varying these two thresholds. The graph shows an interesting relationship between the confidence values of the rules entered into the temporary database and the confidence values of rules that should be checked by the database. The graph shows that the optimal yield is not produced by entering and checking *all* classifications, as was

done in experiment 2. This is not surprising, as such a configuration risks the contamination of the database by poor classifications and hazards misclassification when the classifier is very certain of its initial hypothesis. This data implies that better results can be obtained with the Bayesian classifier as well, and will be examined in future research.

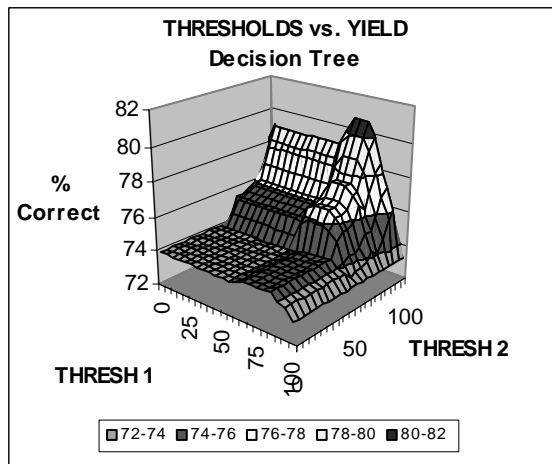


Figure 6. Effects of thresholds on Yield

The thresholds here have been determined based on the *total* yield optimization. That is, while the above thresholds give the highest total yield, there are alternative settings that can increase the yield of specific sub-categories above that which is reported (although to the detriment of other classifications). Finally, it can be observed that the lowest points on the graph are still above the 73.2% accuracy reported for C4.5 alone. This is a result of giving the default classification a confidence level of -1. In this chart, even at 0 for THRESH 1 and 2, the default classifications are put in memory and checked against memory, yielding slightly higher results than C4.5 alone.

Experiment 3: Human Competence

Figure 7 displays the results of the human subjects on location classification. From the results it is clear that classification of location names is not a trivial task. Aside from a poor performance, subjects report that the task is extremely tedious.

It should be noted that subjects were presented with sentences exactly like the example sentences given in the introduction. Further, the subjects were told that the encrypted names were created using a consistent substitution cipher and that they should take advantage of this information. The results from the subjects showed that they did realize that names were repeated and classified them accordingly.

| Class | Subj. 1 | | Subj. 2 | | Subj. 3 | | Avg. % Correct |
|--------------|---------------------|-------------|---------------------|-------------|---------------------|-------------|----------------|
| | # Correct / # Total | % Correct | # Correct / # Total | % Correct | # Correct / # Total | % Correct | |
| City | 5 / 7 | 71.4 | 5 / 7 | 71.4 | 4 / 7 | 57.1 | 66.7 |
| Country | 27 / 37 | 73.0 | 25 / 37 | 67.6 | 17 / 37 | 45.9 | 62.2 |
| Street | 2 / 5 | 40.0 | 2 / 5 | 40.0 | 1 / 5 | 20.0 | 33.3 |
| Region | 4 / 8 | 50.0 | 4 / 8 | 50.0 | 3 / 8 | 37.5 | 45.8 |
| Water | 0 / 3 | 0.0 | 0 / 3 | 00.0 | 0 / 3 | 00.0 | 00.0 |
| Artifacts | 2 / 2 | 100.0 | 0 / 2 | 00.0 | 2 / 2 | 100.0 | 66.7 |
| Territory | 8 / 11 | 72.7 | 5 / 11 | 45.5 | 10 / 11 | 90.9 | 69.7 |
| Mount | 1 / 2 | 50.0 | 0 / 2 | 00.0 | 0 / 2 | 00.0 | 16.7 |
| Total | 49 / 75 | 65.3 | 40 / 75 | 53.3 | 40 / 75 | 53.3 | 57.3 |

Figure 7. Overall Results of Human Subjects

4. Related Work

Our work builds upon the Identifinder system (Bikel et al., 1999), by seeking to expand its general categorization “location” into the eight subtypes described here. While much research has gone into the coarse categorization of named entities, we are not aware of much previous work using learning algorithms to perform the fine-grained classification examined here.

Wacholder et al. (1997) use hand-written rules and knowledge bases to classify proper names into broad categories such as locations, people and organizations. While their work does not focus on fine-grained classification, they employ an aggregation of classification method similar to MemRun. Their method, however, does not use multiple thresholds to increase accuracy.

MacDonald (1993) also uses hand-written rules for coarse named entity categorization. However, where Wacholder et al. use evidence internal to the entity name, MacDonald employs local context to aid in categorization. Such hand-written heuristic rules based on both internal and external evidence are similar to those automatically generated by our learning algorithms.

In categorizing named entities, Paik et al. (1993) attempt a finer classification of location names using manually created heuristics and databases. While their results are strong for closed class categories, such as Countries, their performance suffers on such open class categories as Regions. This is an expected weakness of systems that rely on lists of names for categorization.

Bechet et al. (2000) use a decision tree algorithm to classify unknown proper names (i.e., names not in the lexicon) into the semantic categories: first name, last name, country, town, and organization. While this work is concerned with a slightly finer

semantic distinction than that used in *IdentiFinder*, it is still a much coarser distinction than that focused on in this research. Further, Bechet et al. focused only on those proper names embedded in complex noun phrases (NPs), using only elements in the NP as its feature set. This distinguishes their work from that presented here, in that our algorithms do not depend on the context given by such complex NPs.

5. Conclusions

This experiment produces some interesting information. First, that attempting to classify location names from isolated sentences is a particularly difficult task for human subjects. Second, that Bayesian classifiers can get up to human standards, when augmented with *MemRun*'s ability to store information about previously seen instances. Finally, a decision tree approach augmented with *MemRun* provides a classifier that performs remarkably better than human subjects. At this point, it is interesting to note that Bayesian classifiers assume a level of independence of features that is not assumed by *C4.5*. Thus, it is not remarkable that *C4.5* outperforms the Bayesian classifier, for it does appear that the features chosen in these experiments are very much dependent on each other for proper classification.

One area that requires more research, however, is the investigation of the most appropriate learning algorithm. The complicated nature of the feature interplay when determining classifications suggest that investigation of Hidden Markov Models and Maximum Entropy approaches may be appropriate.

Another area that is still open to investigation is the selection of features. An initial attempt to expand the window of features used in Experiment 1 and 2 did not yield any interesting results, but rather led to data fragmentation. However, wider windows should be looked into. Also possible is the creation of a feature that looks within the single word names of location instances; for example, "Harrisburg" and "Charlottesville" can all be seen as cities.

Features other than word frequency could be examined as well. For example, part of speech tags or thematic roles of the instance may yield some effect. Also, taking advantage of an electronic ontology such as *WordNet* by searching hypernyms of surrounding words for clues as to a name's classification could be advantageous.

Finally, an interesting problem that has not been addressed thus far is the situation in which

orthographically identical instances bear different classifications (such as New York the city, and New York the state). This situation is ignored by *MemRun* as it stands, which can only give a single classification to an instance. One solution to this problem lies in adjusting *MemRun* such that it takes into account the classification that each instance receives in situ along with the average classification it receives throughout the text. That is to say, instead of setting an arbitrary THRESH 2 to check against the classification hypothesis, the program would simply compare the probability of each instance hypothesis against the average probability found in the first round. Only when the probability of the average is greater than that of the hypothesis itself, would the average be substituted. This approach is being attempted in future research.

As of the time of publication, we have run the system over 500,000 articles in the TREC 9 corpus in order to extract instances of locations for import into an ontology. Further, work has begun to extend our research to include the subcategorization of people by career.

6. References

1. Bechet, F., Nasr, A., Genet, F. 2000. Tagging unknown proper names using decision trees. *ACL*, Hong Kong.
2. Bikel, D., Schwartz, R., Weischedel, R. 1999. An algorithm that learns what's in a name. *Machine Learning Special Issue on NL Learning*, 34, 1-3.
3. Jurafsky, D. and Martin, J. 2000. *Speech and Language Processing*. Prentice Hall. Upper Saddle River, NJ.
4. MacDonald D.D., 1993. Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev and J. Pustejovsky, eds., *Corpus Processing for Lexical Acquisition*, pp. 61-76, MIT Press, Cambridge, Mass.
5. Paik W., Liddy, E., Yu, E. McKenna, M., 1993. Categorization and standardizing proper nouns for efficient information retrieval. In B. Boguraev and J. Pustejovsky, eds., *Corpus Processing for Lexical Acquisition*, pp. 44-54, MIT Press, Cambridge, Mass.
6. Wacholder, N., Ravin, Y., Choi, M., 1997. Disambiguation of Proper Names in Text. In *Proc. of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C.
7. Quinlan, J.R., 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers. San Mateo, CA.