Named Entity Classification

(CS224N Final Project)

Chioma Osondu & **Wei Wei**
cosondu@stanford.edu wwei1@stanford.edu

1. Introduction

In this paper, we evaluate the performance of several classification methods applied to the problem of classifying different named entities. Named entity recognition (NER) is defined as a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

While it is possible to do the locating and classification of the predefined categories as the text is parsed, in the discussions that follow, we assume that the first half of named entity recognition has been completed. That is, we assume that named entity recognition has been split into two separate tasks; first, that of locating the atomic elements in text and second, that of classifying the atomic elements retrieved from the text. We assume that the task of locating the atomic elements has been completed and therefore, focus on classifying these atomic elements.

We will discuss three different classification methods namely: decision trees, multinomial naïve bayes, and support vector machines (henceforth, SVM's.) These classification methods were investigated as possible enhancements to the maximum entropy classifier built in assignment #2.

2. Data

We used the data for maximum entropy classification from assignment #2. All the data in the corpus was labeled and belonged to one of five categories / classes, namely: drug, person, place, movie or company. Table 2.1 below shows a sample of the data in the corpus used.

Drug	Person	Place	Movie	Company
Antiflex	Che Guevara	Belfast	102 Dalmatians	Ionics Inc.
Anergan 50	Chris Galvin	Belize	Bon plan	Johnson & Johnson
Apo-ASEN	Dan Andersson	Juan les Pins	Blossoms of Fire	Keyspan Corp

Table 2.1 Sample of the atomic elements in the corpus.

There were a total of 23,122 atomic elements in the training data while the test data contained 2861 atomic elements. Table 2.2 below shows the distribution of the data by category for both training and test data.

	Training Data	**Test Data**
Category / Class	***Number of Atomic Elements***	***Number of Atomic Elements***
Drug	5471	712
Person	4253	493
Place	3762	437
Movie	6909	876
Company	2727	344
Total Number of Atomic Elements	23122	2862

Table 2.2 Distribution of training and test data by category.

3. Classification Methods

From assignment #2, we implemented a Maximum Entropy Classifier to classify different kind of nouns. In this section however, we investigate other classification methods as well as different feature representations.

3.1. Decision Trees

Since decision trees are binary classifiers, we built a decision tree for each category. Given an atomic element, *a*, each decision tree returns a probability that *a* belongs to the class it was trained to classify. The class of the tree with the maximum probability is then assigned to *a*.

Each node in the decision tree is a feature, and is chosen in such a way that it maximizes the Information Gain (IG) according to the ID3 algorithm. The information gain is computed as follows:

$$IG = H(parent) - p * H(child1) - n * H(child2)$$

In the formula, function *H* is called the entropy function. *p* represents the number of atomic elements having some feature, *f*, and *n* represents the number of atomic elements lacking feature *f*. The depth of the tree is another parameter that needs to be optimized. If the decision tree is too deep, it might overfit the data, and therefore generate bad results.

3.1.1. Features and Representation

We tried five sets of features, namely Unigrams, Bigrams, Trigrams, Quadrigrams (Four-grams), and Keywords. Unigrams here refer to single characters. It should also be noted that bigrams here refer to two characters that occur in sequence as opposed to two words (which would be the normal interpretation.) This also applies to trigrams, which are made up of three letters in sequence, and quadrigrams, which are four letters in sequence. The keywords refer to features that were selected by inspection. We basically tried to see what the most obvious features were for each of the categories. Some of the keywords include, corporation, incorporated, management, enterprise, limited, strength, et cetera. All but the last word in the previous list were found to be indicative of the company category (class). The last word, strength, was found to be indicative of the drug class in most cases.

Another set of features used was the number of words occurring in the atomic element. If a particular atomic element contained a single word, its SINGLEWORD feature would be flagged. Similarly, DOUBLEWORD and TRIPLEWORD features. Any atomic element with containing more than five words was said to be verbose, and its VERBOSE feature would be flagged. This was from the observation that the movie and company classes were the most likely classes to have atomic elements with more than four words. Place and person were almost immediately ruled out. Finally, we also had a feature indicating whether or not the atomic element contains a number. This is stored in the HASNUM feature and is useful for identifying drugs.

3.2. Multinomial Naïve Bayes (MNB)

It should be noted that in the implementation of Naïve Bayes, the data was down-sampled. 2000 randomly selected atomic elements per category were used for training, yielding a total of 10000 atomic elements. This was done for two main reasons. First, it was desirable to have equal priors for the different categories. Since the source of the data (i.e. the text from which the atomic elements were pulled) was not known, there was no particular reason to bias the classification towards any particular class. Secondly, most of the categories contain proper nouns; therefore reducing the number of atomic elements dramatically reduced the vocabulary (dictionary) size from about 24,000 to about 15,000 distinct word types. There were two classification approaches to MNB used here and they are discussed below.

3.2.1. Features and Representation

We represent each atomic element as a feature vector whose length is equal to the number of words in the dictionary. Specifically, if an atomic element, *a,* contains the *i*-th word of the dictionary, then we set a_i = *numOccurences*, otherwise, $a_i = 0$. *numOccurrences* refers to the number of times the *i*-th token appears in *a*, regardless of the position of the *i*-th token within *a*. Clearly, we make the MNB assumption that the distribution according to which a word is generated does not depend on its position within *a*.

3.2.2. Stemming and Smoothing

The dictionary used contained all the distinct word types observed in both the training and test data. As a preprocessing step, all word types were stemmed using the Porter Stemming Algorithm[1]. This was done to normalize the word types before they were added to the dictionary. After stemming, word types like "water", "waters", and "water." observed in the data would all become "water".

Notice also that for most NLP tasks, the stop (noise) words like "of" and "the" are removed or de-emphasized, to make the vocabulary size and number of tokens manageable. In this classification task, however, these high-frequency words actually play a crucial role in determining the categories of different atomic elements. For example, it would be very unusual to see the word "of" or "and" in a person's name, but it is highly likely that these same words would exist in the movie names. They are therefore retained in the dictionary, for these reasons.

To ensure that the probability of observing some word type not seen during training is not zero, we used Laplace Smoothing.

3.2.3. Classification (First Approach)

This approach focuses on multi-class classification. We tried to predict the class of each atomic element in the test data by picking whichever of the five classes (drug, person, place, movie or company) has the highest posterior probability.

3.2.4. Classification (Second Approach)

This approach does a one-against-many classification. We changed the labels for the atomic elements belonging to one class to one. All other labels were set to zero. This was done for both the training and test sets. For example, assume we are trying to classify drugs and we just want to be able to tell whether particular atomic elements are drugs or not. Then we set all the labels for the data in the drug category to 1 and the labels for all other atomic elements to zero. To predict whether or not a test datum is a drug, we simply pick the class with the higher posterior probability.

3.3. Support Vector Machines (SVM's)

The features used for the SVM's were the same as those used for Naïve Bayes classification. The method of classification was similar to that of the second approach in our Naïve Bayes implementation. In particular, we trained five separate SVM's because we needed to be able to do one-against-many classification and we had five classes. As in the Naïve Bayes one-against-many classification problem, the labels for the class we wanted to be able to distinguish were set to +1. The labels for all other classes were set to –1.

We assume that the training set is linearly separable; i.e., that it is possible to separate the positive and negative examples using some separating hyperplane. The SVM essentially tries to find a decision boundary that maximizes the (geometric) margin, between the positive and negative examples. The bigger the margin, the more confidence we can associate with the predictions made by the SVM.

We used SVM^{light}, which is Thorsten Joachims'[2] implementation of SVM's. All of the SVM's we trained used a linear kernel function. Details on the specifics of SVM's in general and the particular implementation used in SVM^{light} can be found in the reference at the end of this paper.

4. Results & Analysis

The accuracies reported are simply measures of the number of atomic elements in the test data classified correctly compared to the total number of atomic elements in the test data, i.e.

$$\text{Accuracy} = \frac{\text{number of correctly classified atomic elements in test data}}{\text{number of atomic elements in test data}}$$

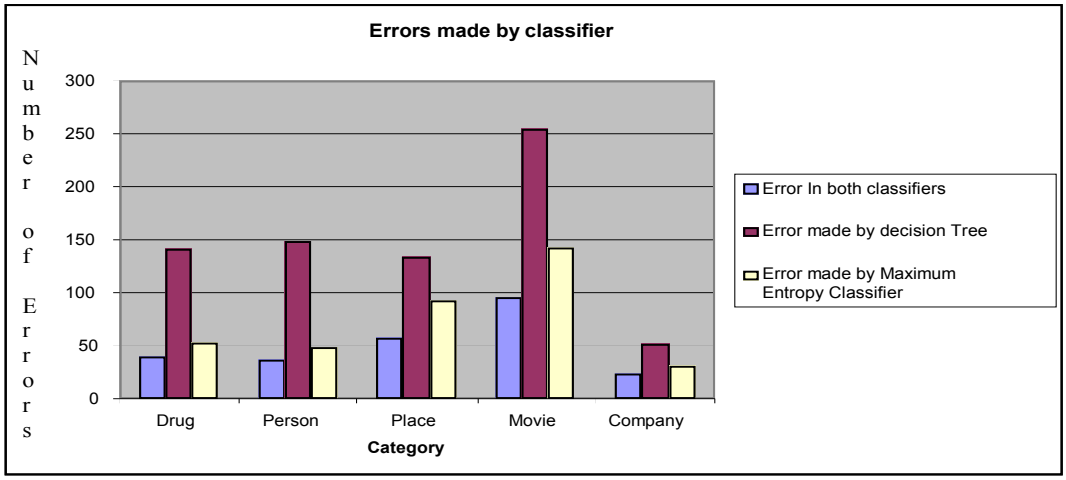
4.1. Decision Trees

The intuition behind the decision tree is that usually, with a single feature, we can pretty much decide whether or not an atomic element belongs to a certain category. For example, given the word "Inc.", we can immediately classify an atomic element as a company (unless of course, this is the atomic element "Monsters, Inc.", in which case, it would be a movie.) As shown in Table 4.1.1 below, once we have the four-gram "inc," there is a high degree of confidence that it is a company name, and the four-gram "corp" is the second most frequent feature, which is also a strong indicator that the name is a company. However, "Animated Corpse" is a movie, it also contains the four-gram "corp", but here "corp" has nothing to do with "corporation".

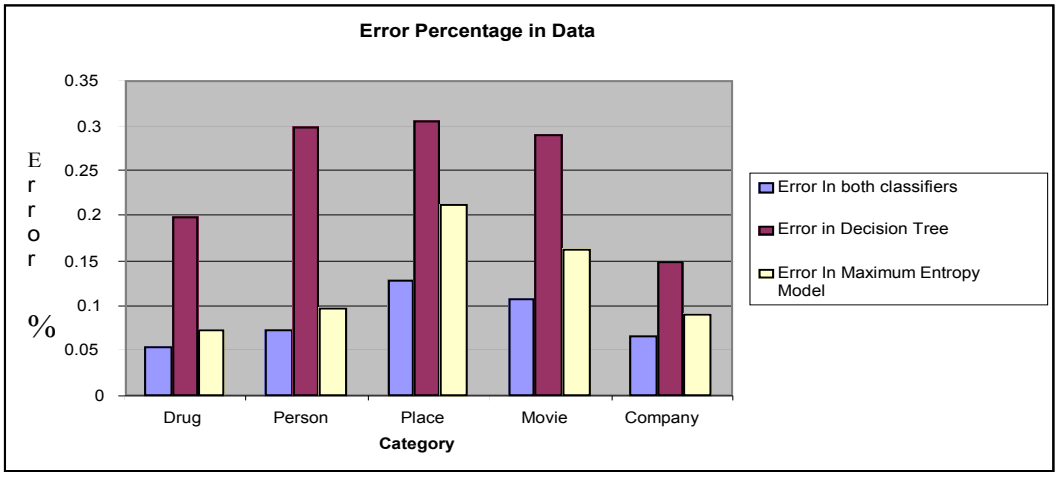
```
Label: company
QUAD- Inc,     P = 0.9974003466204506,    N = 0.07174397960577229
  TRI-gry,    P = 0.0,     N = 0.9982653946227233
    TRI-gry,     P = 0.0,     N = 0.0
    TRI-cid,      P = 0.0,    N = 0.9991319444444444
      TRI-cid,     P = 0.0,    N = 0.0
      TRI-loo,      P = 0.0,      N = 1.0
        TRI-loo,     P = 0.0, N = 0.0
        TRI-loo,     P = 1.0,        N = 1.0
  QUAD-Corp,        P =  0.9964093357271095,      N = 0.0476879962634283
    TRI-rps,         P =  0.0,       N = 1.0
      TRI-rps,       P =  0.0,       N = 0.0
      TRI-rps,         P = 1.0,        N = 1.0
    BI- C,         P =  0.24751439037153322,        N = 0.028104005333606852
      TRI-Co.,       P = 1.0,        N = 0.19843924191750278
        TRI-Co.,         P = 1.0,       N = 1.0
        QUAD-Cap,       P =  0.9791666666666666,       N = 0.15429917550058891
      QUAD-rust,          P =  0.9647058823529412,        N = 0.024003296590089627
        BI- T,          P =  0.9876543209876543,       N = 0.5
        TRI- Lt,        P =  0.9824561403508771,      N = 0.021180968125226015
```

Table 4.1.1. Decision Tree for Company category.

We used different decision tree depths ranging from 1 to 63. The graph below in Figure 4.1.1 shows the performance of the trees at different depths. Here, we did two experiments: the first uses all the features mentioned in section 3.1.1; the other uses all the features except the unigrams. Below, we see that the optimal depth of the tree with all the features is 45, with 74 percent accuracy. Without unigram features, optimal tree depth is 56 with 76 percent accuracy.
We also compared the decision tree with the maximum entropy model (MEM) from assignment #2, with the same set of features. Those results are shown in Figures 4.1.1 (a) and (b) below.



(a)



(b)

Figures 4.1.1 Errors in the classifiers. (Out of 2862 test instances)

Out of the 2862 test examples, the decision tree of depth 45 classifies 599 data instances differently from MEM. Out of the 599 disputed classifications, MEM had 481 correct, and the decision tree had 118 correct. This means, we could probably improve accuracy by combining them (some form of AdaBoast.) In Figure 4.1.1(a), in terms of absolute numbers, we can see the classifiers made the most mistakes in movies. One reason is, the movies have the most diverse vocabulary. For example, "Cecilia" is a person's name, but in the test data, it is a movie. Also, "Frank Herbert's Dune" contains a person's name, so this confuses the classifier as well. In Figure 4.1.1(b), the highest percentage error (the number of error in a category divide by the number of test instance in that category) is in places, because we don't have many features for places. The percentage of the error that the MEM makes when it mistakes a place for movie, or a movie for a place is 74.8 percent of total number of errors made by MEM. This means that there are a lot of similar features between places and movies. Decision Trees made about the same number of errors between places and movies, however, it made many other errors, so mistaking a place for a movie and vice versa make up 20% of all errors made by the decision tree.

There are 249 test examples that both classifiers misclassify, out of which, in 66% of the cases, both classifiers made the same mistake, i.e. classified them into exactly the same category, but the wrong category.

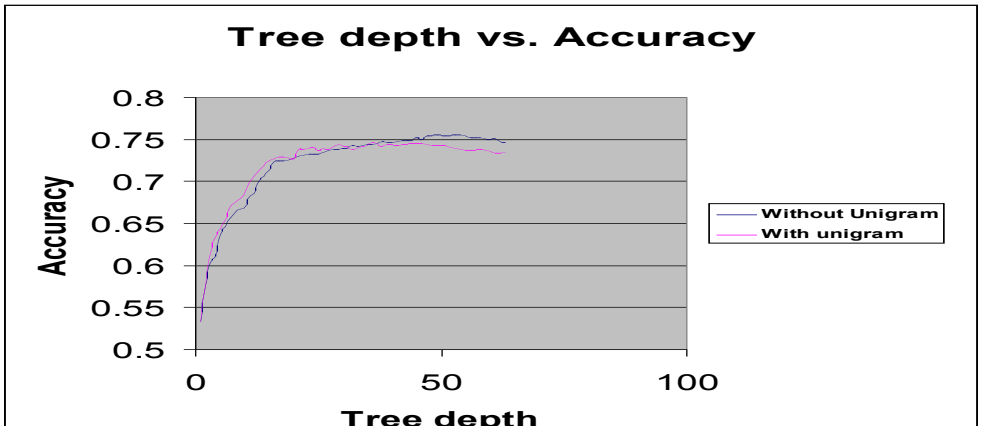


Figure 4.1.2. Accuracy vs. Tree Depth.

There are two possible ways of improving the decision tree we can explore. One is to find better features; the other is to improve the classifier. In a lot of the examples that the decision tree classified wrong, the correct category was usually the second most probable. And the probability that the atomic element belongs to the two most probable categories was usually around 50%. This probably means that the features present were not distinguishing enough and the decision tree could not decide which of the two categories the atomic element belonged to.

For example, features like Corporation and Limited help to classify companies, but they do not appear very often in the training data. Because they are infrequent, the ID3 algorithm did not choose these features as tree nodes since they can only classify a couple of words, and the categories of the words without these features remains unknown. One way to solve this problem would be to collect all features like "Inc.", "Corp.", and "Limited" into a single feature.

Figure 4.1.2 shows the effect of overfitting, in that the accuracy starts decreasing beyond some tree depth. This clearly means that we cannot have a tree of indefinite depth. It also means that adding more features does not necessarily mean gains in accuracy.

It is a bit surprising that taking out all the Unigram features, improves accuracy. Because unigram features like "**:**" help classify movies. The reason the unigrams are not very beneficial is probably the fact that most of the Unigram features do not give a lot of information. For example, all the letters of the alphabet can appear in every one of the five categories used here. A few symbols may yield some information gain, and would therefore be selected by ID3 algorithm but not nearly as many as the number of unigrams that are noisy.

4.2. Multinomial Naïve Bayes

These results show that the Naïve Bayes model performed remarkably well, given its simplicity and ease of implementation.

Classification Method	Accuracy (%)
Naïve Bayes classifier (multi-class classification)	69.21
Naïve Bayes classifier (drug against all other classes)	86.65
Naïve Bayes classifier (person against all other classes)	93.46
Naïve Bayes classifier (place against all other classes)	73.89
Naïve Bayes classifier (movie against all other classes)	87.07
Naïve Bayes classifier (company against all other classes)	97.34

Table 4.2.1. Accuracies for the different Naïve Bayes classification methods.

The multi-class Naïve Bayes classifier was the worst performing algorithm from the results above. The charts below show the predictions of multi-class Naïve Bayes classifier for the five different classes.

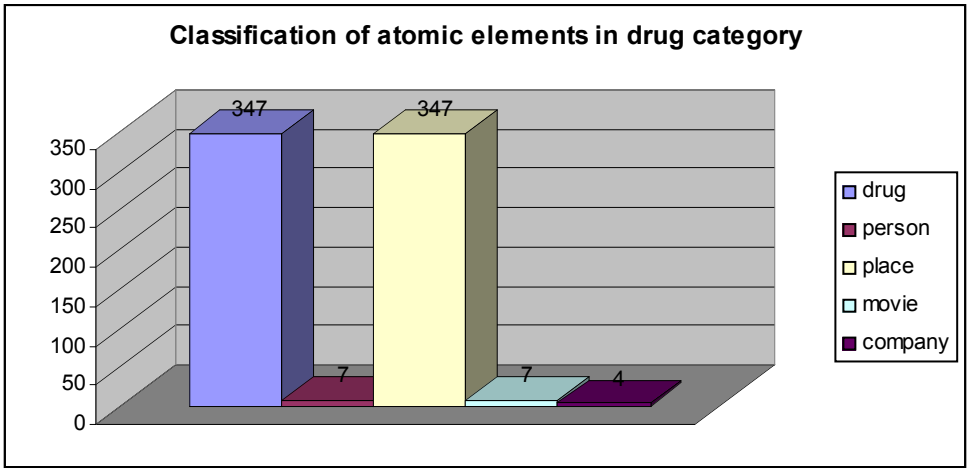


Figure 4.2.1. Predicted classes for atomic elements that actually belong to the drug category.

From the figure above, it can be seen that the multi-class Naïve Bayes classifier does a very poor job of distinguishing between drugs and places. A drug is just as likely to be a drug as it is a place according to the chart above. This result is not really surprising, since the drug and place categories did not contain any particularly distinguishing words (as shown in Table 4.2.2 below) and could just as easily be confused with each other. Some of the misclassified drugs include Actigall, Actisite and Alesse, which were classified as places and those do seem like they could be names of places. Albert Glyburide was misclassified as a person, possibly because the first word in the drug name is Albert, which is in fact a person's name. Mylanta Gas was misclassified as a company, while both Predair A and Aquasol A were misclassified as movies, presumably because of the word "a" which is common in movie names.

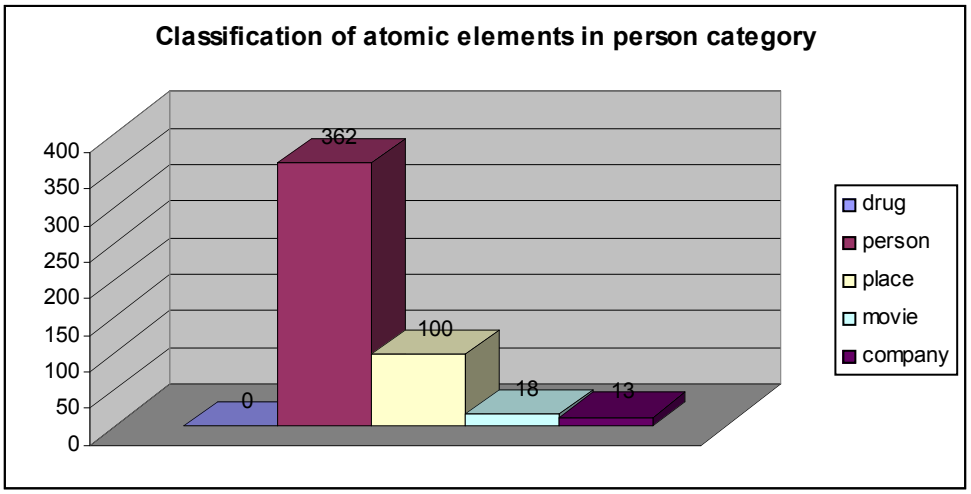


Figure 4.2.2. Predicted classes for atomic elements that actually belong to the person category.

From the chart above, it can be seen that the multi-class Naïve Bayes classifier does a better job predicting atomic elements in the person class than the drug class. There seems to be lingering confusion as to whether an atomic element is a person or a place, but this is not surprising, because places are sometimes named after people. These results show that atomic elements from

the person class are very rarely confused with company names or names of movies. Clearly, this is not always true because certain company names like "Johnson & Johnson" or movie names like "George Washington" are certainly names of people. Some of the misclassified atomic elements in the person class include: Bernt Holmboe misclassified as place, Billy Graham misclassified as a movie and Chevy Chase misclassified as company. Apt, wouldn't you say!

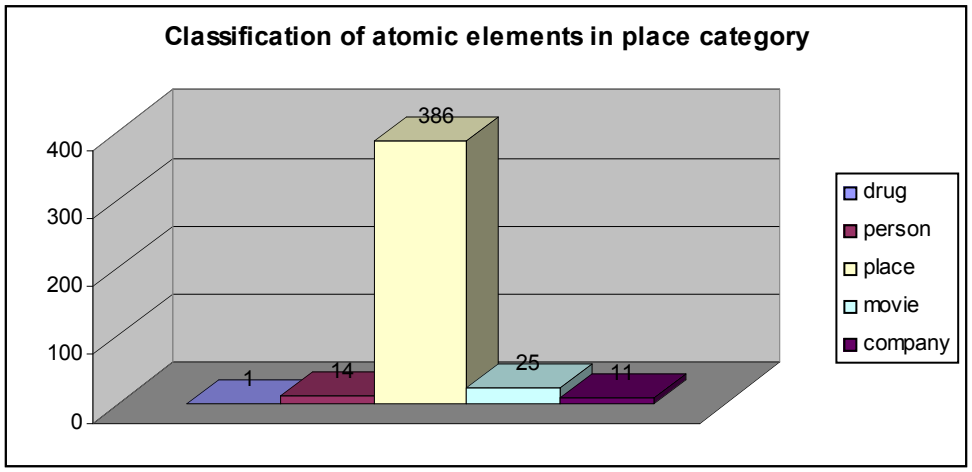


Figure 4.2.3. Predicted classes for atomic elements that actually belong to the place category.

Now, the results of Figure 4.2.3 are surprising! In light of Figure 4.2.1, one would expect that more drugs would be mislabeled as places. This behavior is unexpected because names of certain drugs are just as rare as names of certain places in both the training and test data, and the two classes are expected to be indistinguishable. The multi-class Naïve Bayes classifier would be expected to pick the drug class in these instances, just as often as it picks the place class. However, the results in Figure 4.3 show that the classifier is able to distinguish, for these atomic elements, between the drug and place categories. The only explanation we have is that this is an artifact of the particular data set used. It would be interesting to compare this result on other data sets. Some of the misclassified examples include: Wrightington Bar, the only example misclassified as a drug (possibly because of the word "bar"), Lorraine was misclassified as a person (a very logical assumption, by the way), Crillon le Brave was misclassified as a movie and Arcadia was misclassified as a company. These classifications are not completely unreasonable.

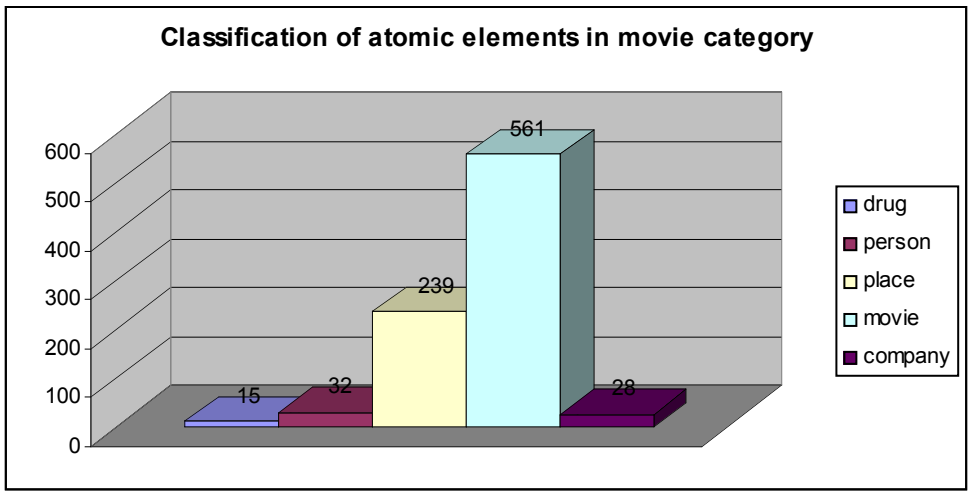


Figure 4.2.4. Predicted classes for atomic elements that actually belong to the movie category.

It would seem from the four bar charts (Figures 4.2.1 through 4.2.4) shown so far that the place category can be found within all other classes. It appears to be the most amorphous class, in that, most of the mistakes made by the multi-class Naïve Bayes classifier are centered on classifying atomic elements from other classes as places. This would be consistent with our general impression that it is hard to pick out atomic elements (especially when they are made up of 3 or fewer words) and categorically say that they are not places, without some other form of analysis. In the case of drugs, if the words look chemical, most people can tell that they are drugs of some sort, but words like "seom" which was misclassified as a place. Who's to say? Some of the misclassified movies include: 41 Shots misclassified a drug (not too farfetched, given that one meaning of the word "shot" is related to drugs), Anna was misclassified as a person (another reasonable classification, given that Anna is a person's name), Arvokkaasti was another example misclassified as place, and Entertainment Life was misclassified as a company.

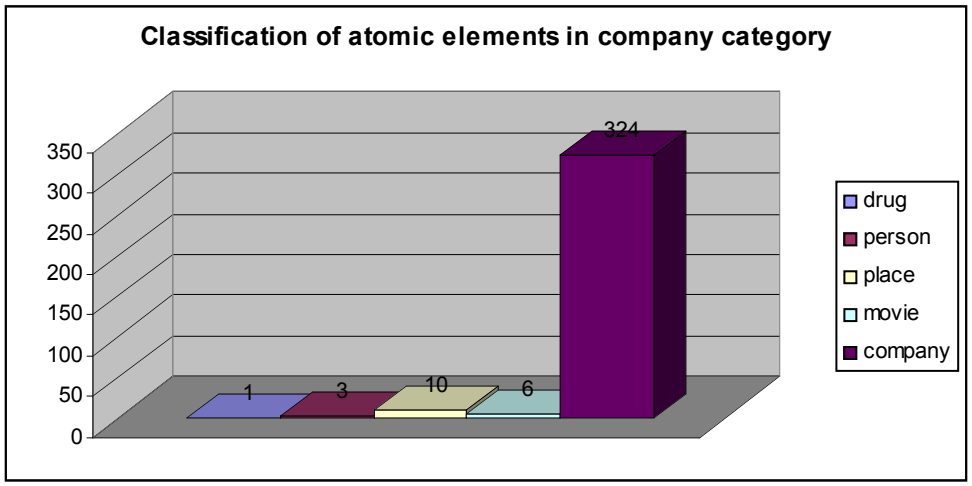


Figure 4.2.5. Predicted classes for atomic elements that actually belong to the company category.

From Figure 4.2.5 above, it can be seen that the multi-class Naïve Bayes classifier does a remarkable job predicting companies. The most mistakes as discussed earlier also arise from classifying certain companies as places. In general, it would seem that it is pretty easy to tell the company class apart from the other four classes. Some of the misclassifications that persisted include: Head N.V. was misclassified as a drug (possibly because of the word "head" as in Sudafed Head and Shoulders), St. Paul Companies was misclassified as a person (presumably because of the words "St." and "Paul", giving the person class a greater posterior probability), Sulzer Medica was misclassified as a place, Copene-Petroquímica do Nordeste was misclassified as a movie.
These observations highlight the fact that there is inherent ambiguity in the classification of different atomic elements into the five categories used in this paper, based solely on words.

Table 4.2.2 below shows ten stemmed words that were found to be the most distinguishing for a given class. They are shown in the order of the most distinguishing to the least distinguishing based on the observed probabilities in Naïve Bayes classification.
In Table 4.2.2 below, the drug column does not seem to show a good distribution of words that one might expect to identify drugs. On closer inspection, though, words like "sugar" and "drowsi" are not foreign to drug descriptions. Words like "metamucil" are definitely reminiscent of drugs and it can almost be said that any reference to "metamucil" would be in association with things medical, hence drugs. The most surprising word in the list is "fleet". The only explanation is that a good number of the drugs in the training data contained the word "fleet", as they would

for drug brand names. In general, sequences of letters (n-grams) in the drug might be better indicators for drug names. Some examples include, "dine", "oxy", "llin", "lax" and other chemical-looking n-grams. The actual words containing these suffixes, and in some cases, prefixes, might be rare (for example, Bisacolax.)

Drug	Person	Place	Movie	Company
plain	john	ashford	i'll	inc
smooth	william	avignon	of	corpor
sugar	st.	bridg	love	trust
tar	sir	de	stori	industri
tum	jame	field	without	fund
drowsi	henri	forest	angel	group
fleet	paul	fort	on	municip
free	thoma	heath	black	compani
hour	robert	mount	man	intern
metamucil	charl	north	is	capit

Table 4.2.2. Most distinguishing words from the different classes.

The column for the person category almost reads like the most common English names for boys, with the exception of "st." and "sir". This is because the training data did not contain many atomic elements for the person category with female names. The female names that did exist in the training data were not nearly as frequent as the male names. The only explanation for "st." and "sir" would be that the particular corpus used here has an unusually high number of occurrences of "st." and "sir". The results for person are unusually remarkable (accuracies over 90%.) This is probably because a lot of the names in both the training and test data are of English origin. It is not difficult to imagine a scenario where the classification of atomic elements as belonging to the person class or some other unknown class would produce dismal results; especially if the names come from different parts of the world.

The words that distinguish the place category are not particularly good indicators of places. A case might be made for words in the list like "fort" and "north" given the fact that quite a few military bases are "Fort"-insert-favorite-fort-here. Also, "north" indicates location and in that sense, could definitely be associated with places. Observe, however, that the word that tops the list is "ashford". This is because there were relatively many places in the training data, containing the word "ashford". Presumably, "ashford" would not be top of the list if one were trying to detect the names of places in, say, Mexico. It must be said, however, that places do not tend to have any particular features that make them easy to identify. The results above seem to emphasize that notion.

In the movie category, some of the words are expected, such as "love", "of", "stori", 'is", and so on. "angel" and "black" seem much more unlikely to be in the list of ten words that are indicative of the movie class. We observed that a lot of movies in the training data contained the word "black". This is clearly a case of overfitting since it is logical to assume that given a different corpus where the word "black" appears in the person class (names like Black, Blackwell et cetera), those atomic elements might be misclassified as movies.

The company category seemed to have the most predictable words. This is borne out by the fact that both the Naïve Bayes classifier and the SVM were able to detect companies with a high degree of accuracy (over 97%.) Also, the words that distinguish company seem to be words that would easily exist in data sets other than the one used in these experiments. Other high-scoring words for companies, not included in the list above, were "limited", "plc" and so on.

As expected, the one-against-many classifications perform better than the multi-class classification. This is because there is more potential for error in the multi-class classification, since the probability of being right is 1 in 5 (20%); whereas, in the one-against-many classification, the probability of making the correct prediction is half (50%). Also, a lot of the errors shown in the charts in Figures 4.2.1 through 4.2.5 would no longer be errors, since four out of the five classes have been merged.

4.3. Support Vector Machines (SVM's)

The SVM model does a pretty good job classifying places, which was the one class that the other classifiers seemed to choke on. The results below, however, do not show a clear winning strategy for classification, given that the one-against-many Naïve Bayes classifier does better than the SVM's for some categories like movie.

Classification Method	Accuracy (%)
Support Vector Machine (drug against all other classes)	84.90
Support Vector Machine (person against all other classes)	91.54
Support Vector Machine (place against all other classes)	85.08
Support Vector Machine (movie against all other classes)	83.23
Support Vector Machine (company against all other classes)	98.74

Table 4.3.1. Accuracies for the different SVM's.

For the sake of brevity, we do not include a thorough analysis of the examples that are misclassified by the SVM's. Some examples (similar to Naïve Bayes) are shown below.

- **Drug class**: Actigall was misclassified as not a drug (possibly because it looks like the name of a place), while North Pole #19, a movie, was misclassified as a drug (possibly because of the number in it.)
- **Person class**: D'alembert was misclassified as not a person's name (possibly because it looks like the name of a place), while Danny and Max, a movie, was misclassified as a person (possibly because of the words "Danny" and "Max".)
- **Place class**: Prescot was misclassified as not a place (because this is, in fact, the name of a person in the training data), while Tales of an Island, a movie, was misclassified as a place (possibly because of the word "Island".)
- **Movie class**: Mexico City was misclassified as not a movie (possibly because it looks like the name of a place), while Lucy Lawless, a person's name, was misclassified as a movie (because the word "Lawless" did, in fact, appear in the title of a movie in the training data.)
- **Company class**: Washington Real Estate Investment was misclassified as not a company (possibly because of words like "Washington", which indicates person or place, and "Estate", which indicates place). The only atomic element that was incorrectly classified as a company was Take It to the Limit, a movie. This is almost certainly because it contained the word "Limit"!

5. Future Work

As discussed earlier, the Maximum Entropy model correctly classified some of the words that the decision trees classified wrong. An interesting extension to the work described here might be combining the different classifiers to improve overall accuracy, in a manner similar to boosting (AdaBoost.) This might serve to exploit the strengths of a particular classifier. For instance, when it is necessary to decide whether or not a particular atomic element is a place, the weight given to the prediction from the SVM might be higher than that of the other classifiers. When it is necessary to classify a movie, the weight of the Naïve Bayes classifier might be bigger than that of the other classifiers. A linear combination of the different predictions from the classifiers could then be used as the final guess.

For the decision tree, it would great to be able to combine features with low information gain features into a one big feature with higher information gain. One way to achieve would be to compare the similarity of two features. As described earlier, features like "Inc.", "Ltd.", and "Corp.", which normally appear in company names, could be treated as one feature. This also serves to reduce the number of nodes in the decision tree, hence its efficiency.

No real use was made of validation data. Another enhancement might be the addition of features if they are seen to improve the accuracy of the classifier on validation data. Unfortunately, in the case of the Naïve Bayes classifiers and the SVM's, this would involve testing thousands of features. An efficient automated way of conducting these experiments might prove helpful to the overall process of correctly predicting the class of an atomic element.

Finally, other ways of implementing Naïve Bayes including Complement Naïve Bayes (CNB)[3], Weight-normalized Complement Naïve Bayes (WCNB) [3] and Transformed Weight-normalized Naïve Bayes (TWCNB) [3] were not explored for lack of time and might yield better results, especially for the multi-class classification problem.

6. Conclusions

In general, using only word-level features does not seem to be nearly granular enough to detect differences in categories of nouns. A better way would be to incorporate the word-level features with other features that incorporate information about particular sequences of letters. The trigrams (three-letter-long sequences), for example, would encapsulate information like "oxy" is likely to be a drug, which the word-level features because of their coarseness cannot do.

Related work done by Stephen Patel [4] and Joseph Smarr [4] show that accuracies of 88% and above are consistently achievable using more granular features.

References

1. C.J. van Rijsbergen, S.E. Robertson and M.F. Porter, *New models in probabilistic information retrieval.* London: British Library. (British Library Research and Development Report, no. 5587), 1980.
2. T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.
3. Jason D. M. Rennie, Lawrence Shih and Jaime Teevan. David R. Kargar, *Tackling the Poor Assumptions of Naive Bayes Text Classifiers*. Proceedings of ICML '03.
4. Steven Patel and Joseph Smarr, *Automatic Classification of Previously Unseen Proper Noun Phrases into Semantic Categories Using an N-Gram Letter Model*. CS224 Final Projects from 2001.