Analyzing Methods and Errors of Proper Name Classification

Timothy Liu
University of California at Berkeley
CS294-19
timothytliu@berkeley.com

ABSTRACT

This paper gives an introduction to Proper Name Classifiers beginning with generative classifiers and ending with discriminative classifiers. The purpose of building these classifiers is to analyze the hidden constructs or ideas that cause data text to be accurately classified. Important questions that will be covered are why certain errors do or do not occur from these classifiers, how to handle or correct these errors, and whether classification techniques can be extended to other fields. The final topic highlights an extremely accurate classifier based on feature extraction.

General Terms

Proper Name Classification, Maximum Entropy Models, Confusion Matrices

Keywords

Naive Bayes', Maximum Entropy, Accuracy, Confidence

1. INTRODUCTION

Classification in natural language processing can be divided into two categories: supervised and unsupervised learning where statistical classification and clustering are respectively the techniques used. Supervised classification relies on having external information about the text. This paper uses prior knowledge of the labels to train its Proper Name Classifiers; therefore, how to evaluate the classifier becomes a main concern.

There are three terms in the evaluation of this paper's Proper Name Classifiers, each of which will be discussed further in this paper. These are defined here for ease of reference:

Accuracy - Given a set of data, features, and weights, how many times a datum is labeled correctly divided by the total number of labeled data.

Confidence - Given a set of data, features, and weights, the Classifier's calculated probability of correctly labeling a specific datum.

EX OUTPUT: Success: Y decirte alguna estupidez, por ejemplo, te quiero guess=movie gold=movie confidence=0.9999999999999823
guess is the Classifier's predicted label
gold is the actual datum's label

Class - There are a specific set of labels or classes for the data. These classes are:

$$[drug, person, movie, place, company]$$

2. CLASS-CONDITIONAL N-GRAMS

The first Proper Name Classifiers built are a subset of class-conditional classifiers. A Class-Conditional Classifier is implemented using a Naive Bayesian network where the text data is independent given or conditioning on the class. In these classifiers, N-gram language models are utilized to generate word probabilities that help calculate the ideal class using Bayes' Rule. The key equation in this classifier calculates the probability of the class given the text data using the posterior distribution over previous text.

$$P(c|s) \ \alpha \ P(c) * \prod_{i=1}^{k} P(s_i|s_{i-n+1}^{i-1})$$

There are different methods of modeling language and smoothing techniques for sparsity; however, the goal of text classifiers is to understand why certain errors were produced and whether they or not can be corrected.

2.1 Generative Classifiers

The given baseline classifier is a MostFrequentLabelClassifier that does not utilize a Language Model and always labels data as the most frequent training label. Two ideas to improve this classifier arose: Using words and iterating through a sentence or using characters and iterating through words. Therefore, an N-Character Language Classifier and an N-Word Language Classifier were built.

As the text data is filled with phrases rather than documents, there is an obvious advantage for using characters to classify. As the order increases, higher-order word models generate more unknown words and require high level smoothing techniques, while higher-order character models are not as sparse and accurately depict distinctive character combinations like 'Inc.', '$\grave{A}$ ma' and 'm∅r'. The Unicharacter language model; however, is too general as there is no sense of context when only looking at the number of instances of each character. Therefore, a higher-order character language model would seem appropriate.

Smoothing techniques better reflect the probability of sparse data, but there are extremely few instances where unknown characters would appear. For ease of use linear interpolated additive smoothing was chosen as the technique for these language models.

Table 1: Accuracies between Baseline, Word, and Character Language Models

Model	Accuracy
Baseline	0.306
Uni-Word	0.621
Bi-Word	0.565
Uni-Char	0.597
Bi-Char	0.758
Tri-Char	0.813

Table 2: Uni-Char Confusion Matrix

Uni-char	DRUG	PERSON	MOVIE	PLACE	COMPANY
DRUG	**421**	35	148	38	70
PERSON	15	**254**	164	6	58
MOVIE	66	80	**615**	38	78
PLACE	32	67	147	**160**	31
COMPANY	14	22	47	1	**260**

Table 3: Bi-Char Confusion Matrix

Bi-char	DRUG	PERSON	MOVIE	PLACE	COMPANY
DRUG	**626**	28	41	7	10
PERSON	17	**398**	66	7	5
MOVIE	68	91	**684**	23	11
PLACE	41	97	127	**170**	2
COMPANY	15	11	26	0	**292**

Table 4: Tri-Char Confusion Matrix

Tri-char	DRUG	PERSON	MOVIE	PLACE	COMPANY
DRUG	**660**	15	22	5	10
PERSON	17	**429**	42	4	1
MOVIE	43	71	**708**	22	33
PLACE	34	82	105	**210**	6
COMPANY	5	5	14	0	**320**

Table 5: Uni-Char: Accuracy and Mass vs. Confidence

CONF. LEVELS	ACCURACY	% OF MASS	STANDARD DEVIAT.	MASS * STD DEV.
.15-.24	.35	0.6%	0.106	0.6%
.25-.34	.27	14.7%	0.021	3.1%
.35-.44	.42	19.4%	0.014	2.7%
.45-.54	.53	14.0%	0.021	3.0%
.55-.64	.63	11.4%	0.021	2.4%
.65-.74	.67	8.1%	0.021	1.7%
.75-.84	.77	7.8%	0.021	1.7%
.85-.94	.85	9.3%	0.035	3.3%
.95 +	.94	14.5%	0.042	6.2%

This trend between character and word language models previously addressed can be seen in **Table 1**.

One type of language model has the advantage over the other, but it is crucial to understand how the better model behaves with actual data, looking beyond the simple number of *accuracy*.

2.2 Confusion Matrices

A Confusion Matrix consists of a matrix of counts where the row represents the actual instances of the class and the column represents the predicted instances of the class. Along the diagonals, the correctly predicted and labeled instances are shown. This matrix is used to identify which label pairs are most confused[1].

In **Tables 2, 3, and 4,** three confusion matrices are shown for a uni-character, bi-character, and tri-character language model. The most common label is MOVIE, which causes most things to be labeled as a MOVIE. For uni-character, the most mislabeled pair was mistaking a **person** as a MOVIE. For both bi and tri character, the most mislabeled pair was mistaking a **place** as a MOVIE.

As mentioned before, because there are more instances of characters in movies, there is a higher probability to assign the movie class given common characters. It is hard to differentiate with one character between **person** and MOVIE because there lacks a specific distinctive feature that only the class **person** has. On the other hand, for example the MOVIE label has many instances of special characters from foreign language films, which is just one way to explain why movies themselves are more correctly predicted.

This can be summarized as having a specific label be too general in its defining characteristics. In the second example of most mislabeled pair between **place** and MOVIE, places mainly have common strings of characters that could appear in movies. The most defining feature of place is that it is most often one word long, but with only using posterior distributions about the previous one or two characters, the length has less of an impact than the probability of specific characters appearing. While these errors seem unsolvable, it is helpful to look at why some errors are produced minimally.

The least mislabeled pair in all three matrices are between mistaking a **company** with a PLACE, which even in some cases never occurs. In this instance, the label **company** has very many distinct features within its corpora such as the appearances of specific short and common characters *Co., Co, Corp., Corporation, Inc., Inc, ltd, LLC, etc.* The appearance of the character 'c' far outweighs the appearance of 'c' in place as well as the combinations produced with 'c'. Therefore, the classifier would almost never choose PLACE as a label for **company** because a company is too distinct overall to match with a more general class.

The ease at which some labels are chosen over others can be divided simply into generality versus specificity. The more general a label is the easier it is to mistake other things for it, but makes it less of a chance to guess with. The more

[1]http://nlp.stanford.edu/ manning/papers/emnlp2000.pdf

specific a label is the harder it is to mistakenly predict a different class for it, but makes it more likely to be a guess that is wrong. However, these classifiers do work well despite the errors induced.

2.3 Accuracy versus Confidence

Character language models perform at higher-orders with increasing accuracy up until n > 4 when the accuracy starts to decrease. However, there is another aspect that combined with accuracy gives a more complete picture of how well the classifier is classifying.

The confidence of a guess is simply the probability that the given text was of that specific class. It would make sense that if the classifier is 70% confident, then 70% of the time it would be correct. In **Tables 5, 6, and 7**, the confidence level and corresponding accuracy for the character language models are shown.

To further illustrate the relationship between confidence level and **Figures 1, 2, and 3** make use of four lines: *Accuracies, % of Mass, Standard Deviation(weighted), and Linear Regression for Accuracies.* The first line simply shows the correlation between Accuracy and Confidence level.

The second line shows the confidence level that appears most when testing the data and visually drags the standard deviation in that direction. The third line shows the true standard deviation multiplied by this shift in mass to show that variance is more pronounced where data is more concentrated.

The last line is a fitted linear regression for the accuracies, which should ideally be the $y = x$ line so confidence and accuracy are equal at all times. However, the regression line keeps falling, showing a lower average accuracy. The third and fourth lines also show that the classifier is steadily becoming overconfident.

It is reasonable to correlate higher confidences and higher accuracies. However, the reason why overconfidence occurs especially in the tri-character is that if a specific sequence of characters is seen, the classifier leans heavily towards a class that contains that sequence. This can be seen as a lack of data to accurately represent all sequences of characters in each class. Also in edge cases, specific errors show that the frequency of common characters truly gives no impression to what class the text really belongs to.

For example, the movie 'Mr. Wong' would trick most humans into labeling PERSON while the classifier predicted this correctly. On the other hand, human brains love to pattern match and have enough context and experience to essentially classify almost all the DRUGS and COMPANIES correctly.

Another example of a prone error is that when there are low confidences, the text is less than five characters long. The fewer the characters, the less distinct the characters can be, so the less the classifier can use distinguishing characteristics to inference over.

ERROR: Ceres guess=drug gold=place confidence=0.4298
ERROR: 0.12h guess=drug gold=movie confidence=0.2961
ERROR: YPF guess=place gold=company confidence=0.3589

Figure 1: Uni-char Accuracy, Mass, and Confidence Comparison

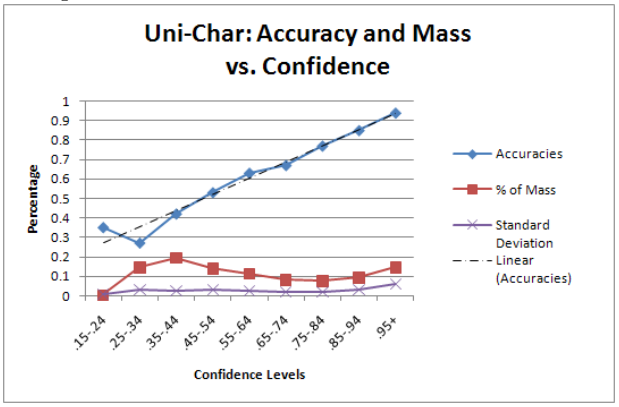


Table 6: Bi-Char: Accuracy and Mass vs. Confidence

CONF. LEVELS	ACCURACY	% OF MASS	STANDARD DEVIAT.	MASS * STD DEV.
.25-.34	.40	0.5%	0.071	0.4%
.35-.44	.37	3.5%	0.021	0.7%
.45-.54	.40	8.1%	0.071	5.7%
.55-.64	.47	7.2%	0.092	6.6%
.65-.74	.50	8.5%	0.141	12.0%
.75-.84	.63	8.1%	0.120	9.7%
.85-.94	.76	12.8%	0.099	12.7%
.95+	.95	51.3%	0.035	18.1%

Figure 2: Bi-char Accuracy, Mass, and Confidence Comparison

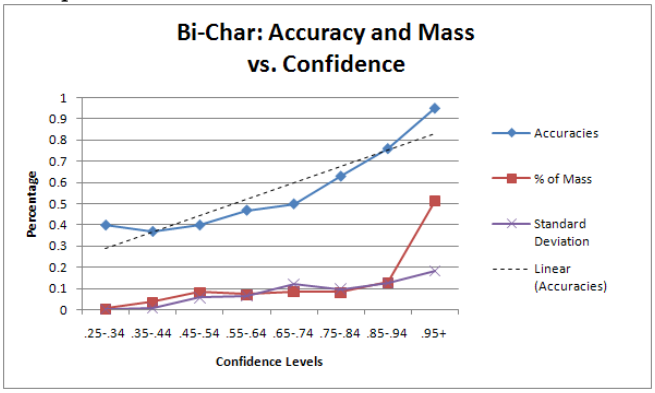
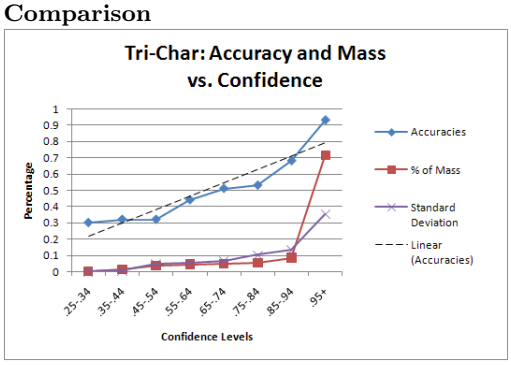


Table 7: Tri-Char: Accuracy and Mass vs. Confidence

CONF. LEVELS	ACCURACY	% OF MASS	STANDARD DEVIAT.	MASS * STD DEV.
.25-.34	.30	0.3%	0.000	0.0%
.35-.44	.32	1.5%	0.057	0.8%
.45-.54	.32	3.7%	0.127	4.7%
.55-.64	.44	4.6%	0.113	5.2%
.65-.74	.51	4.9%	0.134	6.5%
.75-.84	.53	5.3%	0.191	10.1%
.85-.94	.68	18.5%	0.156	13.2%
.95+	.93	71.1%	0.049	35.2%

Figure 3: Tri-char Accuracy, Mass, and Confidence Comparison



3. DISCRIMINATIVE CLASSIFIERS

The second implementation of a Proper Name Classifier was the feature driven maximum entropy classifier. The features describe a discriminative approach to classification to overcome word sense disambiguation. With a discriminative classifier, a multitude of simple and complex features can be combined to accurately classify text.

3.1 Maximum Entropy Model

The main equations for the Maximum Entropy Model are:

$$P(c|d,\lambda) = \frac{e^{\sum_i \lambda_i(c) f_i(d)}}{\sum_{c'} e^{\sum \lambda_i(c') f_i(d)}} \quad (1)$$

$$G(\lambda) = -1 \cdot [\sum_{(c,d)} log\ P(c|d,\lambda)] + \sum_{i,c} \frac{\lambda_i(c)^2}{2\sigma^2} \quad (2)$$

$$\frac{\partial F(\lambda)}{\partial \lambda_i(c)} = -1 \cdot [[\sum_{j:c_j=c} f_i(d_j)] - [\sum_j P(c|d_j,\lambda) f_i(d_j)]] + \frac{\lambda_i(c)}{\sigma^2} \quad (3)$$

The goal of a Maximum Entropy Model is to find the ideal weights for the features, producing the highest probability. **Equation (2)** maximizes the log probabilities generated by **Equation (1)** by adding negative logs. The last term helps to smooth the sparse features by summing weights as the number of features can increase exponentially. The derivative **Equation (3)** with respect to each feature weight helps to optimize the values by checking the difference between the actual and predicted counts.

3.2 Maxent Classifier versus Class-Conditional

After implementing the algorithms using these equations, the baseline Maximum Entropy Classifier had an accuracy of 62.7%, which was better than the class-conditional character classifier. This is most likely because additive smoothing is worse for a 1-char model. However, as seen in **Table 8**, the increasing-order of Maxent Models had lower accuracies than their class-conditional counterparts. The linear interpolation of the n-gram models incorporate multiple contexts that a single feature of the maxent model cannot accomplish.

3.3 Maximum Entropy Features

A substantial roadblock when testing new features was the amount of computational power needed. Therefore, when testing initial features, the baseline was the uni-character model. The first features introduced dealt with recurring errors that would distinguish specific classes from each other. Places, people, and drugs are fairly short while companies and movies have many words. Some errors concerning drugs were all capitalized and errors in companies had phrases like 'Co, Ltd., and Inc.' After adding these features, the uni-char maxent model improved.

Table 8: Accuracies for Maxent Models and the Corresponding Features

Model	Accuracy	Additional Features
Uni-Char	0.627	None
Bi-Char	0.721	None
Tri-Char	0.784	None
Uni-Char	0.648	Num Words, Length < 10
Uni-Char	0.675	Co, Ltd., Inc., ALL CAPS, Hyphen
Linear Inter.	0.794	Weights 1 1 1 - None
Linear Inter.	0.843	Weights .1 .4 1 - None
Linear Inter.	0.821	Weights .1 .4 1 and All Features
Linear Inter.	0.865	Weights .1 .3 .5 .1 - None

Table 9: Optimal Maxent Confidence vs. Accuracy

Level(floored)	.3	.4	.5	.6	.7	.8	.9	1
Accuracy	.57	.49	.58	.63	.79	.87	.91	.99

The next step was to add these features to higher-order models; however, testing individual features would take too long. A solution was found by optimizing the loop structures and multithreading the thousands of function calls. Furthermore, the threads would run on the Millennium clusters at UC Berkeley to cut down the four hours of computation time to twenty minutes for a feature test. The best feature that improved the accuracy to above 80% was linear interpolating the lower-orders like in class-conditional models. This caused the maxent model to be extremely well calibrated, as there was very little overconfidence as seen in **Table 9**. However, errors remained within the lower confidence levels.

ERROR: Lovenox guess=movie gold=drug confidence=0.3969
ERROR: Clare guess=movie gold=place confidence=0.3442
ERROR: Bucet guess=place gold=drug confidence=0.3629

This represents most of the remaining errors because over 65% of the errors were under the confidence level of .65, which is the median confidence level. Furthermore the classifier is more accurate than it is confident around .3. This relates to previously mentioned lack of distinguishing features and low probabilities, showing that perhaps a feature outside of the data is needed like part of speech.

4. CONCLUSION

The approach to Proper Name Classification can be extrapolated to various other topics as the algorithms for learning weights and maximizing the log probabilities can be reused. The Maximum Entropy Model can be useful in Language Identification as part of supervised learning. Features pertaining to language identification that are linear interpolated would most likely prove to be an optimal solution.