Extracting Person Names using Machine Learning Algorithms*

Zihan Zhou, Kirin Hong, Mars Hao

March 10, 2019

Abstract

In this project, we'd like to perform a person name extractor by machine learning methods. We constructed some features of person names in short stories as our model predictors. We performed five models with cross validation to find the best classifier. After identifying the false positive/negative examples, we added black/white list to modify our predictions. Finally, our precision reaches 91% and recall reaches 65%, which both fit the course requirements.

1 Data Source

The documents we obtained are from Chekhov's short stories[1]. We only selected some fragments from each story and got 320 documents in total. In this project, we'd like to extract person names from these documents. For example, we'd like to extract "Otchumyelov" from "The police superintendent Otchumyelov is walking across the market square wearing a new overcoat and carrying a parcel under his arm".

2 Data Process

For each document, we marked the person names as *NAME*, which will cover first name and last name while prefix and suffix are not included. We split our 320 documents into training set **I** with 256 documents and testing set **J** with 64 documents randomly. We only selected words with first character is capitalized. Therefore, those were marked could be treated as positive and others were treated as negative samples automatically. In the training set **I**, we have 1835 positive samples and 1249 negative samples. In the testing set **J**, we 451 positive samples and 316 negative samples. We constructed features for person names based on their previous words and next words by the following rules:

(1) Whether there exists verb among the word. 1 means yes and 0 otherwise.
(2) Whether there exists "'s" in the next word. 1 means yes and 0 otherwise.

*This is a CS 839 Project Stage 1 report in UW-Madison.
[1]https://www.ibiblio.org/eldritch/ac/jr/

(3) Whether there exists prefix in the previous word. 1 means yes and 0 otherwise.
(4) Whether there exists "the" in the previous word. 0 means yes and 1 otherwise.
(5) Whether there exists "of" in the previous word. 1 means yes and 0 otherwise.
(6) Whether there exists "who", "whom" or "whose" in the previous word. 1 means yes and 0 otherwise.

3 Initial Model Selection

After we obtained the above features, we performed models by using decision tree, random forest, support vector machine, linear regression, and logistic regression with 5-fold cross-validation. And we'll evaluate their results to find out the optimal algorithms. The precision, recall and F1 score of the 5 models are displayed in Table 1(The results might be a little different with our jupyter notebook since we used random cross-validation).

Table 1: Cross-validation results of initial models

	Linear Regression	Logistic Regression	Support Vector	Random Forest	Decision Tree
Precision	0.854413	0.738015	0.743693	0.743470	0.743470
Recall	0.401592	0.746216	0.751781	0.751781	0.751509
F1	0.483291	0.733354	0.737585	0.737265	0.737265

From the results above, we found all models perform pretty well except linear regression. SVM performs best among them. After we checked all our models, we can't find more features that could help improve our models. We finally chose decision tree model as our last classifier. The performance of the decision tree model on the testing set **J** are shown in Table 2.

Table 2: Results on testing set J of support vector machine model

	Precision	Recall	F1
Decision Tree	0.8535	0.6718	0.7519

4 Post-processing and Final results

From previous steps, we've already chosen SVM as our optimal model. While our model still can't reach the minimal requirement based on these constructed features and it's hard to improve the model results from debugging. Therefore, we decided to perform post-processing to improve our results.

We trained our decision tree model based on the training set **I**. And we predicted on set **J** to find out false positive/negative samples. In the false positive samples, we noticed some words might have similar patterns as person names. And we created a black list for these words to improve precision. Among the false negative samples, we noticed some person names with prefixes are not identified correctly. So we created a white list for these prefixes to improve recall. We'll predict words contained in the black list as non-person names and words with previous words contained in the white list as person names. Detailed information about black and white list along with other constructed features can be found in our "features.txt".

Finally, we re-predicted the final results on the testing set **J** with rule-based post-processing. The precision, recall and F1 score of our final results are displayed in Table 3.

Table 3: Final Results

	Precision	Recall	F1
Decision Tree	0.9182	0.6718	0.7759