CS839 Stage1 Report

Yunang Chen, Yuanfang Song, Hongqian Xia

1 ENTITY TYPE

We extract **people names** from movie plots. The plots are from Kaggle dataset Wikipedia Movie Plots (https://www.kaggle.com/jrobischon/wikipedia-movie-plots) . Person names in the documents are marked up using … tags. If a name contains multiple words, we marked each word separately. If a name has a possessive suffix ('s), we exclude the suffix from the tag.

Some examples are shown in Table 1:

Table 1 Example Markups

Examples	Text	
1	Raw text	Now that Chris and Frank…
	Marked Text	Now that ** **Chris** ** and ** **Frank** ** …
2	Raw text	Now that Mr. Davy Jones' girlfriend, Rose, has reached....
	Marked Text	Now that Mr. ** **Davy** **Jones** **' girlfriend, ** **Rose** **, has reached ....

2 DATA SET

We label 362 documents with 3292 mentions. After random shuffling, we select the first 200 documents with 1817 mentions as set I and the next 100 documents with 869 mentions as set J.

Table 2 Number of mentioned entities in each document set

	Number of documents	Number of mentions
Total	362	3292
Set I	200	1817
Set J	100	869

3 TRAINING AND CLASSIFIER SELECTION

3.1 INSTANCE GENERATION

We split the sentences into list of tokens, then generate terms by connecting adjacent tokens by length 1, 2, and 3. When a term has multiple tokens, the program marks a term as positive only when all the tokens are marked positive.

An instance contains the current term (of length 1-3 words), the terms before and the terms after.

3.2 FEATURE GENERATION

We generate the following 23 features for each instance:

1. The preceding preposition (on, at, in, ...) if exists.
2. If there is a title (Mr., Dr., ...) before the term.
3. If the previous word is 'named'.
4. The following WH-word (who, which, ...) if exists.
5. If the term is inside parentheses.
6. If the term is followed by a parenthesis.
7. If the term contains common English stopwords (*nltk.corpus.stopwords*).
8. If the term is at the beginning of a sentence.
9. The number of words.
10. The average length of words in a term.
11. If the term contains a title.
12. If the term is followed by a possessive suffix ('s).
13. If the term is preceded by an article (a, the).
14. The previous word's suffix (last two characters).
15. If the term contains an article.
16. If the term is followed by an article.
17. The simplified part-of-speech tag (noun, verb, or other) of the previous word.
18. The simplified part-of-speech tag (noun, verb, or other) of the next word.
19. The simplified part-of-speech tag of the term, if every word has the same tag.
20. If the term contains '.' or '-'.
21. If there is a nearby verb.
22. If there is a nearby 'his' or 'her'.
23. If the term contains a common English word (*nltk.corpus.words*).
24. If every word in the term is a common English word (*nltk.corpus.words*).

3.3 CLASSIFIER M

We perform cross validation on set I to compare the following classifiers:

1. **DecisionTreeClassifier**: Decision tree
2. **RandomForestClassifier**: Random forest
3. **SVC**: Support vector machine with linear kernel
4. **RidgeClassifier**: Linear regression with L2 regularization
5. **LogisticRegression**: Logistic regression with L-BFGS solver
6. **SGDClassifier**: Logistic regression with stochastic gradient descent training
7. **MLPClassifier**: Two-layer perceptron
8. **AdaBoostClassifier**: Adaptive boosting
9. **KNeighborsClassifier**: k-nearest neighbors with $k = 3$

The results are shown in Table 3.

Table 3 Cross-validation Metrics of Classifiers on Set I

Classifier	Precision	Recall	F1
DecisionTreeClassifier	0.870	0.879	0.874
RandomForestClassifier	0.891	0.905	0.898
SVC	0.885	0.945	0.914
RidgeClassifier	0.878	0.953	0.914
LogisticRegression	0.894	0.941	0.917
SGDClassifier	0.921	0.850	0.884
MLPClassifier	0.902	0.927	0.914
AdaBoostClassifier	0.895	0.938	0.916
KNeighborsClassifier	0.877	0.897	0.887

Since the goal is to achieve at least precision of 90% and as high recall as possible, we select MLPClassifier as Classifier M.

3.4 CLASSIFIER X

During the previous cross-validation step, we find that our Classifier M (MLPClassifier) fails to converge to the default tolerance within the default iterations. However, as we increase the maximum number of iterations, the performance metrics slightly drop, which indicates it is possibly prone to overfit. Therefore, we add the early stopping.

After re-doing the cross-validation, MLPClassifier with early stopping still has the highest recall while achieving the baseline precision (shown in Table 4).

Table 4 Cross-validation metrics of Classifier M on Dev Set I after debugging

Classifier M	Precision	Recall	F1
MLPClassifier with early stopping	0.902	0.938	0.919

Therefore, we select MLPClassifier with early stopping as Classifier X and apply it on the test set J. The results are shown in Table 5.

Table 5 Metrics of Classifier X on Test Set J

Classifier X	Precision	Recall	F1
MLPClassifier with early stopping	0.920	0.913	0.916

3.5 POST-PROCESSING RULES/CLASSIFIER Y

Since our result has already achieved the target precision and recall, we do not add any post-processing rules. Therefore, classifier Y is the same as classifier X.