Game Name Identifier

CS838 Project Report – Stage 2

GAURAV MISHRA [gmishra2@wisc.edu]
OM JADHAV [ojadhav@wisc.edu]
PALLAVI MAHESHWARA KAKUNJE [kakunje@wisc.edu]

1. Overview:

The goal of this stage of the project is to extract the important information and identify an entity type from natural language text documents, using a supervised learning approach.
Entity type: Game Application names.
Few examples include games like Mini Metro, Super Cat Tales, Neon Chrome etc.
Text Documents: Reviews of Game Applications.
Given a game review, our trained classifier accurately identifies whether each entity corresponds to name of a game app or not.

2. Entity Tagging:

In this step, we parse all the documents and markup positive and negative entities. All the mentions of game application names are marked with a positive tag. Other names such as publisher name, developer name, character name etc and words starting with uppercase are marked with negative tag.

Total number of documents: 300
Total positive entities: 1151
Total negative entities: 1900

3. Feature Extraction:

We have identified following features:

1. The entity is a stop word.
2. The entity corresponds to maximum mentions in the document.
3. The entity is a noun.
4. The entity is preceded by the words “by”, “from” or “developer”
5. The entity’s next word contains $ (price of the application)
6. All the words in the entity are present in the dictionary.

4. Learning Step:

In this step, we perform cross validation on the training set. Precision and recall is computed and is used to select the best classifier.
The documents are divided into two sets:

Set	**No. of documents**	**No. of positive tags**	**No. of negative tags**
Training Set	200	786	1269
Test Set	100	365	631

4.1. Initial Cross-Validation:

5-fold Cross Validation performed on the training set using multiple classifiers provided the following result:

Classifier	**Precision (%)**	**Recall (%)**	**F1 (%)**
Decision Tree	76.8	71.9	74.269
Support Vector Machine	76.9	71.8	74.262
Random Forest	76.5	72.0	74.182
KNN	50.9	76.5	61.127
Linear Regression	51.2	75.3	60.954
Logistic Regression	75.9	72.1	73.951

Based on the reading obtained above, we observe that Decision Tree provides better precision (76.8), recall (71.9) and F1 score (74.269). Therefore, we chose Decision Tree.

4.2. Cross-Validation After Feature Refinement:

We noticed that the main area of improvement was to identify the difference between human/character/stage names and game names. We added below features to separate them.

1. Count of occurrence in the review document is one such important feature.
2. looking up the dictionary to check if the word is a meaningful English word.

5-fold Cross Validation performed on the training set using multiple classifiers after feature refinement provided the following result:

Classifier	**Precision (%)**	**Recall (%)**	**F1**
Decision Tree	91.33	89.8	90.72
Support Vector Machine	91.05	90.05	90.82
Random Forest	91.33	89.82	0.9057
KNN	90.60	90.84	90.57
Linear Regression	87.91	83.43	85.62
Logistic Regression	90.72	90.71	90.72

Based on the reading obtained above, we observe that Support Vector Machine provides higher F1 (90.82). Therefore, ***Support Vector Machine*** is the classifier which best suits our requirement.

5. Testing

The selected classifier support vector machine is used for testing on the test data.
Finally, in this step we use our learned classifier to predict output for the test set.
The reading looks as follows:
Precision: 91.37 %
Recall: 87.12 %
F1: 89.19 %