Michal Trněný
Masaryk University, 2015

Classification of Steam Games Descriptions

Game descriptions on Steam provided by a developer are supposed to introduce the game. However, they may start with irrelevant information, e.g. storyline. The aim of this study was to train a classifier which would label descriptions based on how informative they are.

Data

The used dataset was obtained by extracting description segments from JSON files downloaded using Steam API. The dataset contains 3,021 games where each game has an "About the game" attribute, describing the game. These descriptions were chosen for classification based on how informative they are, i.e. whether one can easily figure out what kind of game they are viewing. The resulting dataset's only attribute is a plain text.

The texts were cleared off encoding errors (cause unclear, e.g. apostrophe would sometimes display incorrectly, other times correctly - it is possible that different descriptions used different encoding).

Out of the 3,021 descriptions, 200 were randomly chosen for manual labelling by one annotator. The requirement for a description to be informative was that the first 400 characters had to describe the genre, goals of the game etc., not only story for instance.

Evaluation

The evaluation was performed in R, utilizing Weka. 8-fold cross-validation was used. The document-term matrix was limited to 200 words (using Weka's "wordsToKeep" parameter). In order to keep both classes at roughly the same size, negative examples were used twice in the training set in each fold. The test set always contained the same number of positive and negative examples. Naive Bayes Multinomial proved the most effective in early experiments and was hence used further. The table below presents results from the following settings:

- preserving letter case, no stemming
- converting to lower case, no stemming
- preserving letter case, stemming applied
- converting to lower case, stemming applied

Lower case and stemming were used further as they proved useful. The next step was using a 50 word limit instead of 200. Further, negative examples were used 3 and 4 times in the training set.

As a next step, games with less than 10 players in average during the first 2 months after release were filtered out.

Principal Component Analysis (PCA) was used last - with converting to lower case, applying stemming, and using 2x negative examples. The document-term matrix used to create components was limited to 200 words using Weka's "wordsToKeep" parameter. Naive Bayes was unable to work with new values as they contain negative numbers, hence SMO was used. Since SMO generally provides worse results on this data, an experiment was performed with 50 word limit, converting to lower case, and applying stemming for comparison with PCA (using 50 components).

	Accuracy	Precision 1	Recall 1	Fscore 1	Precision 0	Recall 0	Fscore 0	Precision avg	Recall avg	F-score avg
200 limit	0.7	0.68	0.76	0.72	0.74	0.64	0.68	0.71	0.7	0.7
200 limit, lowercase	0.71	0.69	0.78	0.73	0.75	0.64	0.68	0.72	0.71	0.71
200 limit, stemming	0.7	0.68	0.79	0.73	0.75	0.61	0.67	0.71	0.7	0.7
200 limit, stemming, lowercase	0.72	0.68	0.85	0.75	0.81	0.6	0.68	0.75	0.72	0.72
50 limit, stemming, lowercase	0.74	0.7	0.88	0.77	0.84	0.6	0.68	0.77	0.74	0.72
50 limit, stemming, lowercase, 3x negative examples	0.72	0.7	0.81	0.74	0.79	0.64	0.69	0.74	0.72	0.72
50 limit, stemming, lowercase, 4x negative examples	0.73	0.73	0.72	0.72	0.75	0.74	0.73	0.74	0.73	0.73
77 more successful games, 50 limit, stemming, lowercase	0.6	0.57	0.89	0.69	0.78	0.32	0.42	0.68	0.6	0.56
SMO, 50 limit, stemming, lowercase	0.67	0.62	0.94	0.74	0.92	0.39	0.49	0.77	0.67	0.62
SMO, stemming, lowercase, PCA - 50 components	0.69	0.7	0.88	0.78	0.65	0.38	0.48	0.68	0.69	0.67

Conclusion

Naive Bayes Multinomial performed well on the dataset. As experiments showed, converting to lower case and stemming combined improve overall accuracy. While limiting the number of words in document-term matrix to 200 seemed reasonable, further adjustments showed a limit of 50 provides even better results. There was an apparent problem with examples being incorrectly classified as positive which can be balanced by multiplying the negative examples, with an expected trade-off, however.

Taking only the more successful games into consideration does not yield any better results. This could be explained by larger studios having marketing specialists write these descriptions, resulting in texts being more difficult for machine-learning. While Naive Bayes Multinomial could not have been used with PCA, a comparison on SMO suggests that using PCA provides better results than the "wordsToKeep" parameter available in Weka.