Devoir 2

Gustavo Alonso Patino

Abstract— **In this work, we present the use of regular expressions in the extraction of information in a corpus of games. The corpus is composed of comments from various gamers and specifies whether or not the player recommends the game. Various features were obtained using regular expressions such as: different game names, facets, types of games and the names of the most mentioned video game companies. Additionally, we were able to identify important characteristics of each class using the TF-IDF word embedding method, taking advantage of this procedure, we could train 3 machine learning classifiers, obtaining a good performance in all the algorithms studied. Finally, we use a tagging method to identify the most used adjectives in each class, obtaining relevant information about the positive and negative points of the games.**

I. Introduction

One of the most recognized methods in text analysis is the use of regular expressions (RE), which is a language used to specify chain of strings (or characters) in a corpus. This practical language is used in all computer languages, word processors and word processing tools such as Unix, grep, or Emacs tools [1]. In general, a regular expression can be defined as an algebraic notation used to characterize a set of characters or strings that define a search pattern. Typically, they are particularly useful for searching for specific information in texts, ”search and replace” operations in strings, or for the validation of an entry. For this, we must specify a search pattern and apply it to the corpus, finding efficient results even in large-sized corpus. A regular expression used search functions which will inspect through the corpus, returning all texts that match the pattern. The pattern can be a text or a set of characters or numbers. For example, the Unix grep command line tool takes a regular expression and returns each line of the input document that matches the expression. A search can be designed to return each match on a line, or just the first match.

II. Experiments

A. Data pre-processing in text classification

- Cleaning the data: Tokenization, Capitalization, Select only alphabetical tokens, Avoid stop words.
- Word embeddings: TFIDF with unigrams.
- In TFIDF, we define the number of features according to the 10k most viewed words in the corpus.

B. Classification models and hyperparameters

To choose suitable values for the algorithms hyperparameters, we put in a grid several values for the hyperparameters and combine them in an dictionary with the form: grid=Hyperparameters:[values]. Then the GridSearchCV function of sklearn was used to execute a classification task for each cross-value. At follows we show the algorithms used and the best hyperparameters values found in the validation:

- Naive Bayes: Laplace smoothing parameter:0.1
- logistic Regression: Regularization strength C:0.9
- SVM:Linearkernel, penalty parameter of the error term C:0.4

III. Results

We work with a corpus extracted from the platform ”steam” which contains comments from players about various games. Comments can describe a game and specify whether the game is recommended or not by the player. In the first stage, we use regular expressions to try to discover the games described by the players. For the regular expression we use the pattern ”is a game”. In the file ”games.txt” we show the result of the search. In front of each item we identify the text with the label ”true” if the items are really talking about a game, given that the pattern used in the regular expression provided noise results. The regular expression identified 330 comments of which 127 actually corresponded to games. In the file ”list_games.txt” we listed all the games identified in the corpus. In the same manner, we can use the comments to identify different the type of games (see the complete list on the file type_ games.txt) and different facets. In the file ”facets.txt” we show a list of the identified facets, these facets are quite relevant since in many cases they are associated with the final decision of recommending or not on a game, examples of these type of facets are: atmosphere, fashion, colors, story, world development, aesthetic, animation style, strategy, music, scenario, graphics, characters, bugs and storyline.

TABLE I: Performance different classifier using TF-IDF unigram embedding

Model	**F1 score**	**Accuracy**
Naive Bayes	0.71	0.83
Logistic Regression	0.72	0.84
SVM(Linear Kernel)	0.75	0.84

Since the corpus explicitly defines whether or not a player recommends a game, we can do analysis in each of the classes. An important item to be analyzed are the words most used by players who recommend or not a game. For this, we did a TF-IDF decomposition using the 10k most frequent words in the corpus. The TF-IDF method is interesting because it allows us to identify the most frequent words in texts (TF) that at the same time do not appear much in other texts (IDF). In Fig.2 and Fig.1 we can see the results of the most relevant features in texts with label 0 (not recommended game) and with label 1 (recommended game). Certain features shown in the figures could clearly help a classifier to achieve a good performance in a classification task. In Table.I we show the performance of several models applied in the binary classification problem . In the classification task, we used the F1 metric since the positive and negative comments were unbalanced, so that a poor classifier could generate great accuracy. We also note that a unigram model generates good results since the characteristics of the positive or negative comments are usually condensed on very specific words. Naive Bayes had a remarkably good performance. Usually this method behaves well in text classification, additionally, the weight of our classification task falls on very particular words increasing the performance of this method. Similarly, the SVM method with linear kernel had a good performance in the classification, this is due to the fact that probably in the 10k dimensional space of the features, the data presents a good tendency to be linearly separable, this separability is reasonable given that the positive and negative comments have particular features of high TF-IDF score which are not shared between the classes.

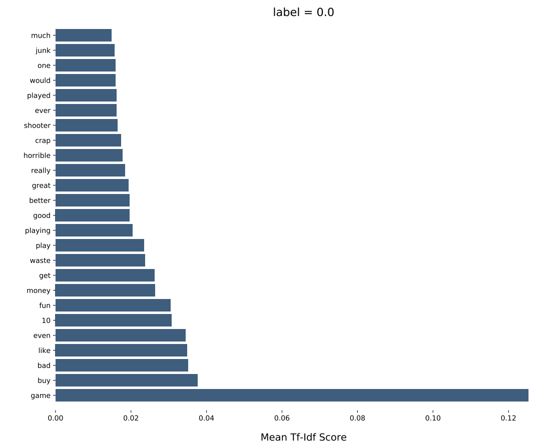

Fig. 1: Most important features found using TF-IDF label 0

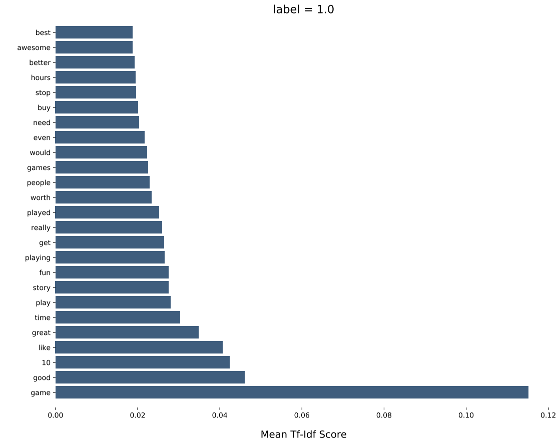

Fig. 2: Most important features found using TF-IDF label 1

Furthermore, we identify that in many cases the non-recommendation comments (Fig.1) are associated with the price, since the word ”money” is a feature with a high score, and the word ”buy” is actually associated with the recommendation ”not to buy a game”. In the same way, we see that ”great” is a word with a fairly high score in label 0, this is because in many comments the players start with positive comments about the idea of conception of the game, but then complain about the final product. On the other hand, in the case of positive recommendations (Fig.2) many of the features with the highest score refer to adjectives used by players to describe the game, in fact in the non-recommendations some adjectives also appear among the features with the highest score. For this reason, we decided to study the most used adjectives in each group. In order to solve this task, we use the ”postag” method of the NLTK library in python to set a tag on the tokens of each corpus text, so if the tag corresponded to the set [”JJ”, ”JJR”, ”JJS”] we define the word as an adjective. In the figures Fig.3 and Fig.4, we see in an illustrative way the most frequent adjectives used by each group. In the case of non-recommendations one of the most used adjective is ”bad”, we can notice that in many cases the players claim of technical aspects of the game such as being broken or too heavy to download, or if the game’s history is flat. In Fig.3, we notice that the most commonly used positive adjective is ”good”, ”great” and ”different”, followed by fairly common expressions such as ”awesome”, ”best” or ”amazing”. Finally, we listed the name of the video game companies cited in the corpus along with the number of times they were cited, the results can be seen in the file ”enterprises.txt” . We had the difficulty of associating the word of the companies directly to a positive or negative label, since in many comments the classification of the game was not directly associated with the company, since the name of the company was mentioned to make a comparison.

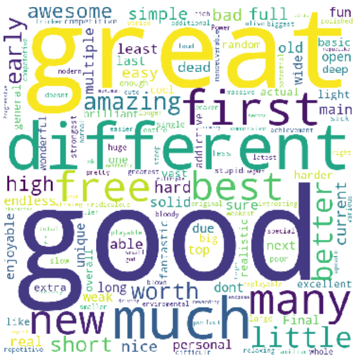

Fig. 3: Most used adjectives in label 1

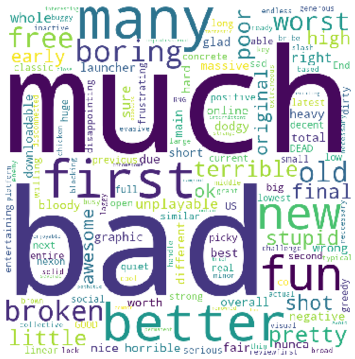

Fig. 4: Most used adjectives in label 0

IV. Conclusion

We used regular expressions to extract relevant information in a corpus composed by comments about games. Using this procedure, it was possible to identify different game names (see games.txt file), facets (facets.txt file), game types (type_ games.txt file) and the names of the most mentioned video game companies (enterprises.txt). The corpus is made up of comments about a game and specifies whether or not the player recommends the game, from this information, we could extract relevant information in each of the classes recommendations: positive (labeled as 1) and negative (labeled as 0). Specifically, we identified the most important features in each class using the TF-IDF word embedding, finding that many of the highest scoring features in each class were exclusive, which could help a machine learning method to have a good performance in the classification of the texts. This premise was corroborated by testing 3 methods: Naive Bayes, Logistic regression and linear SVM, finding that the F1 metric was always greater than 70% and the accuracy even greater than 80%. In the same way, we used a tagging method to identify the most used adjectives in each class, from these adjectives, we infer important results regarding negative and positive points of the game, for example among the negative points we see that the players complain with great frequency of the difficulty downloading the game, the bugs or about the price.

References

[1] Jurafsky,M. Martin,J. Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Third Edition draft. Stanford University.