Natural Language Processing Applied To Sentiment Analysis

– Machine Learning From Theory to Practise – Project –

Johann FAOUZI : johann.faouzi@ensae-paristech.fr
Mélanie FINAS : melanie.finas@ensae-paristech.fr
Hicham JANATI : hicham.janati@ensae-paristech.fr
Peter MARTIGNY : peter.martigny@ensae-paristech.fr

Friday, January 13th 2017

1 Introduction

NLP (Natural Language Processing) is a field at the intersection of computer science, artificial intelligence and linguistics. The aim of NLP is to be able to understand natural language in order to perform some tasks such as translation, named entity recognition, sentiment analysis, question answering... In this project, we will focus on NLP applied to Sentiment Analysis.

Sentiment Analysis has raised a lot of interest recently, especially in the media industry. Its main goal is to identify opinions that are expressed in textual comments. For example, a fashion clothing company could use the textual reviews of its products online to know what the consumers think of them, in order to build better products, adapted to consumers' needs. Another example lies within presidential elections. Today, a candidate needs to take social media into account, as it is a huge source of textual information. Because these information are produced as textual comments, sentiment analysis will be very useful to understand what the citizens think of different candidates and their opinions on different societal issues.

Hence, given the growing number of sources of textual opinions being published on the internet, especially in social media, sentiment analysis has become a major industrial task using machine learning technics. Social media like twitter have even produced their own api to use sentiment analysis (tweepy on Tweeter).

In this project, we will restrict to the tasks of predicting a binary outcome (does the reviewer like the product or not ?) or a multi-class outcome (On a certain scale : how would you rate this product ?).

2 Several paradigms for Sentiment Analysis

When deciding whether a reviewer will like or not a product, there are mainly two approaches to tackle the task of sentiment analysis :

— **Lexicon based approach**
— **Machine Learning methods**

As [LZ12] explain, lexicon approaches are based on dictionaries of words annotated according to their sentiment polarity. Hence, based on a list of positive words and a list of negative words, the task will be to sum all word-scores in the sentence to assign it a sentiment polarity. These methods are said to perform very well on specific domains. However, it is known to be poorly generalizable to other domains. This approach suffers from overfitting.

The second approach focuses on using machine learning methods to predict the polarity of a text. Using regular machine learning algorithms means that we are able to represent the texts as features, ready to be fed to a learning algorithm.

This project explores two main methods to represent text as vectors, in order to carry out a learning algorithm.

— **Bag of Words Model**
— **Word embedding with Word2Vec**

2.1 Bag of Words Model

One of the first attempts to featurize texts dates back to the $50's$ with [Har54]. In his *bag of words model*, the algorithm first learns a vocabulary from all the provided textual data. Then, each text is represented by computing the relative frequencies of its words, both within the document itself and the whole set of documents. This methodology is known to give good results, however it suffers from an important drawback : by considering only the frequencies of the words, we lose the order of the words, which are very informative about the structure of texts and the relations existing between words.

In this project, we will use 2 ways of measuring the importance of a word in a text :

— By counting the number of its appearance in the text (term-frequency)
— By counting both its term-frequency and its rarity between the different texts (Inverse Term Frequency, IDF). Indeed, we want to take into account the fact that a word that appear often in all document may not be very discriminant.

2.2 Word Embedding

In the recent years, there have been a large excitement around the task of embedding words into low-dimension spaces, while still preserving contextual information.
Word2Vec is the model that has become very popular. Based on skip-gram and Continuous bag of words models, introduced in [MSC+13],the goal is to design a deep neural network that learns jointly a usual task (like classification) and the word vectors. The embeddings that result from this training part provide surprisingly high performance score on tasks related to semantic and syntactic metrics. Along with another word embedding algorithm, *Glove*, *Word2Vec* is the state of the art for lots of NLP tasks, including *Sentiment Analysis*.

Once these vectors are learnt, there is still a need for a strategy to represent the whole text as a vector. In this project, we will explore 2 methods using *Word2Vec* word vectors to build text vectors :

— **Vector averaging** : we will take as text vector the average of its word vectors.
— **Clustering** : as we built a natural distance between our word vectors, there is a good chance that groups of words will be significant to predict sentiments. Hence, we will use the K-Means algorithms to groups the words into clusters. Then, we will consider the clusters we created as features. Hence, each text will be represented as a vector of size the number of clusters, and for each cluster we assign the number of words in this cluster.

2.3 Output Predictions Algorithms

In this project, once the representation of the textual data has been learned, our task is to predict if a reviewer will like or not a product. As our measure of "like" or "dislike" is different whether the measure is binary or ordinal, we will use 2 approaches to adress the problem :

— **Regression** : Given a set of reviews with ratings, we will learn models able to predict the output rating of the review. We will test the usual regression algorithms.
— **Classification** : We will arbitrarily say that the most well-rated products will be assigned the label "positive" and the most under-rated products with the label "negative", and then perform classification task on them. We will test the usual classification algorithms.

3 Dataset

We will use the *Amazon book reviews dataset* available on the UCI Machine Learning Repositery website. It consists of reviews of 7 books from readers, with a total of 219 242 reviews. The readers give a rating from 1 (very negative review) to 5 (very positive review).

3.1 Pre-processing

There will be a few pre-processing tasks to be done before building the representation matrices :

— Cleaning : remove html tags,
— Usual NLP pre-processing : lowercase, stopwords, stemming (different for bag of words and Word2Vec).

3.2 Training & Evaluation

We will first train our models on a small subsample of the whole dataset. Hence, as our models have lots of features, there will be computation time issues to follow closely. We need to be cautious while sampling because we have to respect the initial distribution of ratings, in order to have a subsample similar to the full dataset.

In order to evaluate our predictions, we will use usual metrics :

— **Regression** : we will use both the *RMSE* and the *MAE* metrics.
— **Classification** : we will use the accuracy and the F1 measure.

4 Conclusions & further work

In this project, we review some very different methods to predict the sentiment of a review, from bags of words to the deep learning *Word2Vec* model. However, there have been other improvements to the task of sentiment analysis due to deep learning.

For example, we have modeled the sentences starting from the representation of words into vectors, and then we used the word vectors to produce text vectors. In this last process, we lose the ordering of the words, and hence lose some information. Hence, we could learn a representation directly for the sentence level, or even for the paragraph level. This was highly studied in [LM14].

Finally, we could also implement the recursive neural network used in [SPW+13], which is state of the art for many sentiment analysis datasets.

References

[Har54] Zellig S. Harris. Distributional structure. In *Word*, page 156, 1954.

[LM14] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *arXiv :1405.4053*, May 2014.

[LZ12] Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463, 2012.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[SPW+13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October 2013. Association for Computational Linguistics.