Sentiment Analysis Using a Graph Based Representation

Complex and Social Networks

Lorenzo Galizia

24/1/2018

1 Introduction

Expressing opinions and posting reviews about places visited or movies seen has become really popular nowadays, this has led to the need to make sense of this huge amount of data.

Sentiment analysis is the task of identifying whether the opinion expressed in a text is positive, negative or neutral in general, or about a given topic. For example: "Harry Potter is such a good movie, highly recommends 10/10", expresses positive sentiment toward the movie, named Harry Potter, which is considered as the topic of this text.
Sometimes, the task of identifying the exact sentiment is not so clear even for humans, for example in the text: "I'm surprised so many people put Harry Potter in their favorite films ever list, I felt it was a good watch but definitely not that good", the sentiment expressed by the author toward the movie is probably positive, but surely not as good as in the message that was mentioned above. In many other cases, identifying the sentiment of a given text is very difficult for an algorithm, even when it looks easy from a human perspective, for example: "if you haven't seen Harry Potter, you're not worth my time. if you plan to see it, there's hope for you yet."

The human language is complex therefore teaching a machine to analyze the various grammatical nuances, cultural variations, slang and misspellings that occur in reviews provided by users is a difficult process. Teaching a machine to understand how context can reflect tone is even more difficult. Advancements in machine learning and natural language processing techniques made it possible to analyze user reviews and identify the user's opinions towards them. This methods of sentiment analysis are useful in a wide range of domains, such as business or politics.

In the last year, with the exponential growth of the usage of the social media (reviews, forum, discussion, blog, and social network), people and industries use more and more this information as support to their decisional process.

In this project the objective is to detect the sentiment of a set of documents retrieved from Twitter. Twitter is a free platform of social network, and nowadays is one of the social network most used.

Here is proposed an analysis based on the graph structure of the corpus (*tweets*), and it is divided into two part:

- *Experiment* part: in which the performances of the model are tested.
- *Application* part: in which the model is used to investigate on the opinion of the Italian population about the upcoming elections.

The rest of the report is dived into the *Method* section, in which is basically explained how the model was build and how it works, the *Results* section, in which all the results are shown, the *Discussion* section, in which the results are discussed and explained, and the *Conclusion* section.

1.1 Datasets

The experiments reported in this project were carried out in the framework of the SemEval 2015 (Roshental, 2015), Task B. The dataset to **Train** the model is downloaded following the guidelines provided form the coordinators of the workshop, using the public streaming Twitter API to download the *tweets*. It is composed by a set of documents (*tweets*) regarding different topics and labeled as positive, negative or neutral (objective). Instead the **Test** datasets are retrieved from different editions of the SemEval competition, and these are:

- SemEval2014, Task 9, sub task B.
- Semeval2013, Task 2 (tweets).
- SemEval2013, Task 2 (SMS).

All these datasets are composed by four features that are *ID1*, *ID2*, *response* (negative/neutral/positive) and *text* (tweets). The *Table 1* report the dimension of the datasets used for the experimental part.

	rows	columns
SemEval2015	9665	4
SemEval2014 (tweets)	8926	4
SemEval2013 (tweets)	3730	4
SemEval2013 (SMS)	2094	4

Table 1: Dimensions of the "experimental" datasets

For what concern the *application* part, the **Train** data to fit the model was retrieved from the evaluation campaign Evalita 2014 (Basile, 2014). There are two main components of the data: a generic and a political collection. The latter has been extracted exploiting specific keywords and hashtags marking political topics, while the former is composed of random tweets on any topic.
The model fitted with this data is used to classify new tweets, which are been downloaded directly through the Twitter API searching specific words about the three main parties/coalitions of the politics of Italy. The queries are composed from the name of the leaders and the principal names of these groups:

- *Centro Sinistra*: (Renzi, PD, Gentiloni, partitodemocratico, centrosinistra)
- *Movimento 5 Stelle*: (Grillo, DiMaio, M5S, movimentocinquestelle)
- *Centro Destra*: (forzaitalia, fratelliditalia, leganord, Meloni, Salvini, centrodestra)

In order to provide impartial results, for each topic are downloaded only 7000 *tweets* which are stored in a single dataset (*Table 2*).

	rows	columns
Evalita 2014	4513	4
Politics of Italy	21002	4

Table 2: Dimensions of the "application" datasets

2 Method

The sentiment analysis process used in this project, showed in *Figure 1* (Castillo, 2015), is composed by two principal phases: **Training phase** and **Testing phase**.

2.1 Training Phase

This step mainly consist in the preprocessing of the raw *tweets*, construction of the graph and the evaluation of the words (vertices) through the usage of centrality measures.

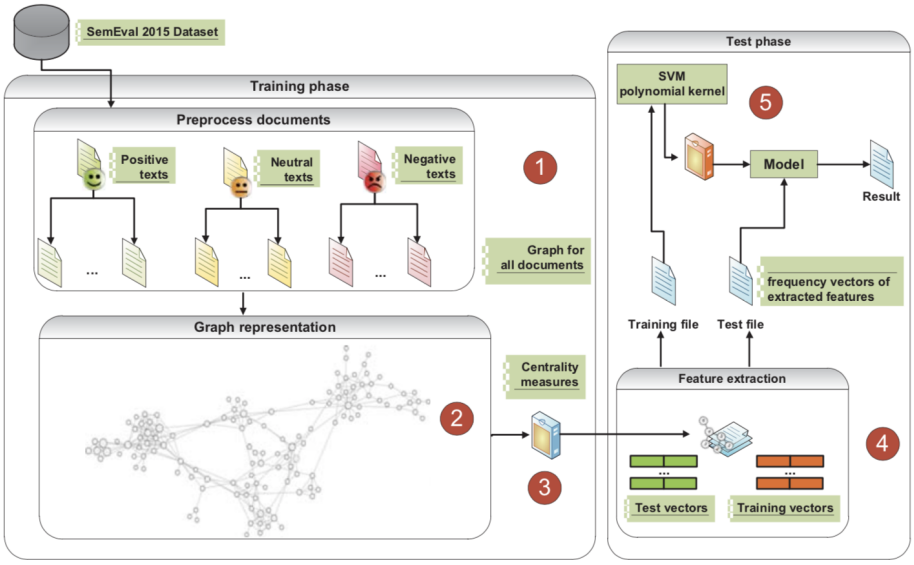

Figure 1: Sentiment Analysis Process.

2.1.1 Text Preprocessing

Twitter users often create their own words and spelling shortcuts, punctuation, misspellings, slang, new words, URLs, genre specific terminology and abbreviations, so is necessary to pre-process this raw data to extract and retrieve informative words.
In this project the *Natural Language Tool Kit* (NLTK), available on Python, is used on the data to create the documents vocabulary.
The pre-processing operations applied to the data are:

1) *Basic preprocessing operations*: make all the documents in lower case and remove all the non-alphanumeric digit;
2) *Stopwords removal*: all the stop words are removed with particular attention to common terms in *tweets* (e.g. "RT"/Re-Tweet);
3) *Stemming process*: reducing inflected (or sometimes derived) words to their word stem, in order to consider all synonymous all the words with the same stem.
4) *Normalization process*: removal of accent and punctuation.

2.1.2 Graph Representation of Text

Text documents can be represented as a graph in many ways. The **co-occurrence of words** (Sonawane, 2014) is an effective way to represent the relationship of one term over another one in texts.
In this project the relationship between two words is attributed to the presence of these into a sliding window of size N (between 2 and 10) on each documents and following this basic idea the co-occurrence was built.
In particular if a term is new in the text then a node is added to the graph and an undirected edge is added if this term co-occur with another terms within a certain window size.
The co-occurrence graph presented in this project is weighted and in order to weight the undirected edges is used a function that assign a tag to a pair of associated terms equal to the total time that this two terms co-occur within a window of size N in all the documents in the corpus.

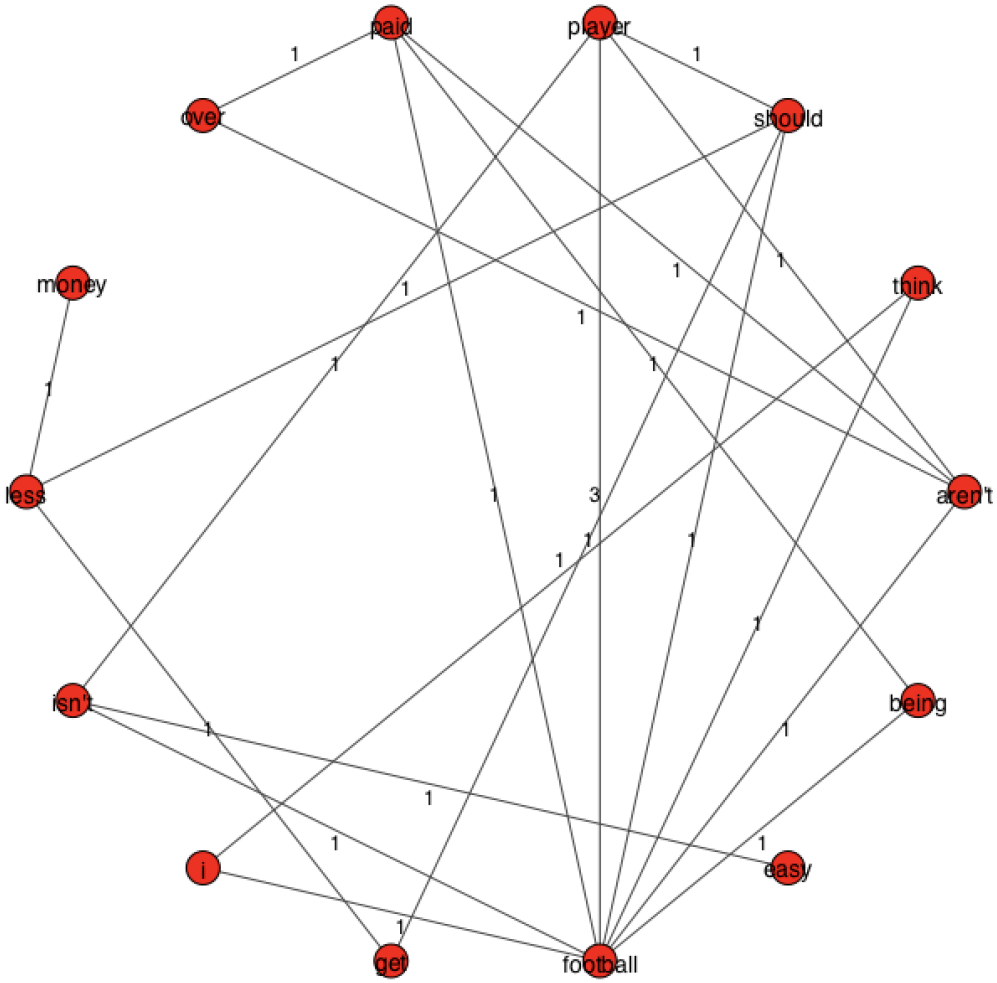

Figure 2: Example Graph.

As an example consider the two sentence, which represents *tweets* with a related topic, already preprocessed:

- S_1: "football player aren't over paid be football player isn't easy".
- S_2: "i think football player should get less money".

In the *Figure 2* is shown the co-occurrence graph for S_1 and S_2, built with a window of size 3 and following the guide line explained above.

It is important to notice how the shift of the windows, in a typical co-occurrence graph, begin in the last term of previous window, but as an addition in this project is proposed also a window that shifts on the next word in the sentence, providing so an highly connected representation of the corpus, and the results of this new approach are discussed and compared with the classic one in the next sections.

2.1.3 Graph Based Analysis

The main objective of using the co-occurrence graph for sentiment analysis is to retrieve the most informative words into the corpus, and so the most important vertices in the co-occurrence graph.
The centrality measures have the main purpose to do this within the graph and the measures take into account in this project are:

- **Degree centrality** (DG): number of links incident upon a node

$$DG(i) = k(i)$$

- **Closeness centrality** (CG): average length of the shortest path between the node and all other nodes in the graph

$$CG(i) = \frac{n-1}{\sum_{j \neq i} d(i,j)}$$

  a node is important if is close to everybody else, thus the more central a node is the closer it is to all other nodes.
- **Betweeness centrality** (BC): number of times a node acts as a bridge along the shortest path between two other nodes, so a node is important if it lies in many shortest-paths

$$BC(i) = \sum_{j<k} \frac{g_{jk}(i)}{g_{jk}}$$

  , g_{jk} is the number of shortest-paths between j and k, and $g_{jk}(i)$ is the number of shortest-paths through i.
- **Eigenvalues centrality** (EC): assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question and the centrality of this node is proportional to the sum of scores of its neighbors

$$EC(i) \propto \sum_{j} A_{ij} EC(j)$$

The most important vertices within the corpus are extracted as the top 100 ranked vertices for each centrality measures, avoiding duplicates.
After multiple tests the usage of all 4 measures seems to retrieve the most informative features to have better results in the testing phase.

2.2 Testing Phase

After found the best words within the corpus, this unstructured data need to be converted into meaningful data in order to apply machine learning algorithms.
After the pre-processing step, data are converted to numerical vectors where each vector corresponds to a *tweets* and entries of each vector represent the presence of feature in that particular *tweets*, which features are the words selected in the section 2.1.3, corresponding to the most important vertices in the co-occurrence graph.
Then the next steps consist in the vectorization of the categorical data retrieved from the *tweets* and in the fit of the model.

2.2.1 Feature Extraction

The main problem at this point of the process is that we have only nominal (categorical) data, that are represented as strings, and we have to interpret this data in order to find the importance of each features (words) in each class of documents.
So due to this purpose, a frequency of occurrence vector (Manning, 2008) is built and after that the data are vectorized and ready to be passed to the model.
The vectorization of textual data to numerical vector is done using *Term Frequency - Inverse Document frequency (Tf-Idf)*.
The *Tf-Idf* score is helpful in balancing the weight between most frequent or general words and less commonly used words.
Term Frequency calculates the frequency of each token in the review, then this frequency is multiplied by

frequency of that token in the whole corpus. *Tf-Idf* value shows the importance of a token to a document in the corpus.
Considering word j and document i the *Tf-Idf* for this pair document-word is calculated as

$$w_{j,i} = tf_{j,i} \times idf_j$$

where the idf_j id the inverse *Tf-Idf* frequency of word j across all documents,

$$idf_j = log_2 \frac{|D|}{|\{document \in D | j \in document\}|}$$

which is the logarithm of the total number of documents divided by the number of documents that contain word j.
Tf-Idf assigns higher weights to words that are less frequent across documents and, at the same time, have higher frequencies within the document they are used. This guarantees that words with high *Tf-Idf* values can be used as representative examples of the documents they belong to and also, that stop words, which are common in all documents, are assigned smaller weights.
It is important to notice, since we are transforming at the same time both the **Train** and the **Test**, that the *Idf* scores, corresponding of the frequency of a given term in the whole corpus, calculated in the *Train* are used also for the words in the *Test* data in order to make realistic the assumption that the *Test* data are brand new data to classify.

2.2.2 Model

Once both the *Train* and the *Test* data are vectorized, these can be used respectively to fit the model and to observed the results.
In order to classify the data a SVM classifier is chosen and in the *Table 3* is shown the hyper parameters for this model.

	hyper parameters
Kernel	'rbf', 'linear', 'poly'
Gamma	0.125, 0.5, 2, 8
C	2, 8, 32

Table 3: Hyper parameter SVM

All the possible combinations of these parameters were tested by means of a Grid Search 10-folds Cross Validation and the optimal one was chosen taking into account one measure:

- **Matthews correlation coefficient**, a correlation coefficient between the observed and predicted binary classifications that returns a value between -1 and $+1$ retrieved with the formula

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

  coefficient of $+1$ represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation.

As a last step the best model is taken and used to predict the *Test* data.

2.2.2.1 Support Vector Classifier (SVC)

Support vector machines (SVM) are machine learning algorithms that analyze data for classification and regression analysis.
Their remarkably robust performance with respect to sparse and noisy data is making them the system of choice in a number of applications from text categorization to protein function prediction. When used for

classification, they separate a given set of binary labeled training data with a hyper-plane that is maximally distant from them (known as *the maximal margin hyper-plane*).
Given a training set of document-label pairs $(x_i, y_i), \quad i = 1, ..., l$ where $x_i \in R^n$ and $y \in \{1, 0\}^l$, the support vector machines require the solution of the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

$$s.t. \quad y_i(\mathbf{w}^T\phi(x_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0$$

$C > 0$ is the penalty parameter of the error term.
The C hyper-parameter tells the algorithm how to balance the two competing objectives which are to maximize the margin between the two classes and to not allow any samples to be misclassified. If $C = 0$ then the algorithm does not allow any samples to be misclassified. If your data is not linearly separable then the algorithm will not be able to find a separating hyper plane. If $C > 0$ then the algorithm can trade-off some misclassified samples in-order to find a margin that better separates the remaining points.
For cases in which no linear separation is possible, they can work in combination with the technique of *kernels*, that automatically realizes a non-linear mapping to a feature space. The hyper-plane found by the SVM in feature space corresponds to a nonlinear decision boundary in the input space.
Given two objects, the kernel outputs some similarity score. The objects can be anything starting from two integers, two real valued vectors to trees, provided that the kernel function knows how to compare them.
The simplest example is the **Linear kernel**, also called dot-product. Given two vectors, the similarity is the length of the projection of one vector on another :

$$k(x, y) = x^T y + c$$

The **Polynomial kernel** is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

$$k(x, y) = (\alpha x^T y + c)^d$$

Adjustable parameters are the slope α, the constant term c and the polynomial degree d.
Another interesting kernel examples is the **Gaussian kernel** (radial basis function kernel, or RBF kernel). Given two vectors, the similarity will diminish with the radius of σ. The distance between two objects is "reweighted" by this radius parameter:

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right)$$

The adjustable parameter σ plays a major role in the performance of the kernel, and is tuned carefully to the problem at hand.

3 Results

As mentioned before, in this project are proposed two studies, so this section regarding the results are divided into two section:

- *Experimental* section: reports the results on the datasets retrieved form the SemEval workshops, and with the main aim of show how this graph based model work on twitter data.

- *Application* section: reports the results obtained from the analysis of the sentiment analysis on a real topic, the Italian elections.

3.1 Experimental Results

In this section are shown the results of the model, fitted with the **SemEval 2015** dataset, on different datasets. For each datasets the precision, the recall, and the F1-score are reported with the confusion matrix, the normalized accuracy and the relative Matthews correlation coefficient.
In addition is reported also the measure proposed in *SemEval 2015* (Rosenthal, 2014):

$$\frac{F1_{pos} + F1_{neg}}{2}$$

3.1.1 SemeEval 2014

	precision	recall	f1-score	support
negative	0.40	0.05	0.09	1287
neutral	0.55	0.90	0.68	3438
positive	0.72	0.45	0.55	2827
avg/total	0.59	0.58	0.53	7552

Table 4: Results 2014

	Negative	Neutral	Positive
Negative	68	1045	174
Neutral	41	3079	318
Positive	61	1500	1266

Table 5: Confusion Matrix 2014, True X Predicted

- **Normalized Accuracy** = 0.5843
- **Matthews Correlation Coefficient** = 0.3127
- **SemEval Measure** = $\frac{F1_{pos}+F1_{neg}}{2}$ = 0.32

3.1.2 SemeEval 2013 (tweets)

	precision	recall	f1-score	support
negative	0.41	0.05	0.08	403
neutral	0.56	0.92	0.70	1270
positive	0.78	0.47	0.59	1115
avg/total	0.63	0.61	0.56	2788

Table 6: Results 2013 (tweets)

	Negative	Neutral	Positive
Negative	19	334	50
Neutral	11	1163	96
Positive	16	575	524

Table 7: Confusion Matrix 2013 (tweets), True X Predicted

- **Normalized Accuracy** = 0.612

- **Matthews Correlation Coefficient** = 0.358
- **SemEval Measure** = $\frac{F1_{pos}+F1_{neg}}{2}$ = 0.33

3.1.3 SemeEval 2013 (SMS)

	precision	recall	f1-score	support
negative	0.38	0.05	0.08	394
neutral	0.68	0.86	0.76	1208
positive	0.59	0.62	0.61	492
avg/total	0.60	0.65	0.60	2094

Table 8: Results 2013 (SMS)

	Negative	Neutral	Positive
Negative	18	316	60
Neutral	14	1042	152
Positive	16	169	307

Table 9: Confusion Matrix 2013 (SMS), True X Predicted

- **Normalized Accuracy** = 0.653
- **Matthews Correlation Coefficient** = 0.35
- **SemEval Measure** = $\frac{F1_{pos}+F1_{neg}}{2}$ = 0.345

3.1.4 Case Window 1-step

The results shown here are of the graph model constructed with the atypical window with 1-step, and evaluated on the *SemEval 2014* dataset.

	precision	recall	f1-score	support
negative	0.32	0.03	0.05	1287
neutral	0.54	0.88	0.67	3438
positive	0.69	0.44	0.54	2827
avg/total	0.56	0.57	0.52	7552

Table 10: Results 2013 (SMS)

	Negative	Neutral	Positive
Negative	38	1029	220
Neutral	42	3041	355
Positive	37	1533	1257

Table 11: Confusion Matrix 2013 (SMS), True X Predicted

- **Normalized Accuracy** = 0.574
- **Matthews Correlation Coefficient** = 0.289
- **SemEval Measure** = $\frac{F1_{pos}+F1_{neg}}{2}$ = 0.29

3.2 Application Results

Here the results on the more recent tweets about the politics of Italy are reported. Like before the precision, the recall, and the F1-score are reported with the confusion matrix, the normalized accuracy and the relative Matthews correlation coefficient but this time, since the lack of classified *Test* data, these results are related to the *Train* dataset (*SemEval 2015*).

	precision	recall	f1-score	support
negative	0.56	0.69	0.62	1312
neutral	0.65	0.61	0.63	1301
positive	0.62	0.48	0.54	1022
avg/total	0.61	0.60	0.60	3635

Table 12: Results 2013 (SMS)

	Negative	Neutral	Positive
Negative	910	241	161
Neutral	364	798	139
Positive	344	191	487

Table 13: Confusion Matrix 2013 (SMS)

- **Normalized Accuracy** = 0.6038
- **Matthews Correlation Coefficient** = 0.401

For what concern the *Test* data here below are reported the percentages negative, positive or neutral of the tweets for each party:

	Movimento 5 Stelle	**Centro Sinistra**	**Centro Destra**
Negative	37.39	51.29	68.27
Positive	19.37	15.12	5.46
Neutral	43.04	33.6	26.27

Table 14: Percentage of sentiment by party.

4 Discussion

The purpose of this project was to investigate the impact of a graph based approach in a sentiment analysis on twitter data.

The first conclusion to be drawn is the evident difficulty of the model in detecting the negative documents. Indeed it is possible to see in all the results, on the *Tests* dataset, that almost the 80% of the negative *tweets* are misclassified as neutral. This result could be due to the strong unbalanced nature of the *Train* dataset (*Table 15*) and this could be the cause of the lower presence of words relative to negative documents in the most retrieved most important features.

	Negative	Neutral	Positive
SemEval 2015	1000	3476	2665

Table 15: Count values of the SemEval 2015 dataset.

On the other hand the results on the positive and neutral classes are sufficiently good, indeed in average the 50% of the positive class and the 88% of the neutral class are correctly classified.

For what concern the graph construction, during the project, was analyzed the opportunity to work with a model from an highly connected graph. Indeed it was tried a different window sliding more slowly to transform the texts from string to graph, in particular was tested the 1-step window. Anyway this approach does not give us good result (*Table 10/11*), and this could be expected since doing this the weight of the edges increase exponentially and in some cases we could lose information about important word (e.g. brokerage vertices).

In the application part is applied the model on Italian tweets. In this case the three classes are equally balanced through the document of the *Train* dataset (Evalita 2014), as we can see in the *Table 16*, so we can expect fair results on the *Test* data after deduction of normal error that has to be take under consideration. The results of **Movimento 5 Stelle** are in line with the last polls, in fact seems the party with the minor percentage of negative tweets, instead the results between **Centro Sinistra** and **Centro Destra** seems different form the surveys, that show the coalition of Berlusconi rising again and a fall of the Renzi's **Partito Democratico**.
What can be seen form this conflicting result is that Berlusconi's candidancy in the *centro destra* coalition and Salvini's aggresive politics did not achieve a good resonance of the Twitter users.

5 Conclusion

In this project it is presented an approach that uses a supervised learning method with a graph based representation.
The results show that this model in a balanced situation can provide good classification, but there is still a great deal to improve on the SemEval 2014 dataset.
Anyway this approach can be very useful, against the classic one, in fact a graph based analysis does not required detailed linguistic knowledge, domain or language specific collection and the most informative features are retrieved avoiding to deal with the enormous amount of data produced by the n-grams approach for example.
In order to improve this model as future work I would propose:

- build a personalized list of stop words with respect to the training dataset, that is shown be an effective procedure (Schofield, 2017);
- try different supervised models (e.g. Neural Networks) to deal with multiclass classification.
- try different way to retrieved the most important vertices in the graph, in order to obtain more precise features for the model.

6 References

Rosenthal S., Nakov P., Kiritchenko S., Mohammad S., Ritter A., Stoyanov V. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter, Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, Colorado, USA.

V. Basile, A. Bolioli, M. Nissim, V. Patti, P. Rosso 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task.

E. Castillo, O. Cervantes, D. Vilarino, D. Baez and A. Sanchez, UDLAP: Sentiment Analysis Using a Graph Based Representation, 2105. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 556–560, Denver, Colorado, June 4-5,2015 © 2015 Association for Computational Linguistic.

Sonawane S and Kulkarni P. 2014. Graph based Rep- resentation and Analysis of Text Document: A Survey of Techniques. Journal of Computer Applications, 96(19):1-8.

Manning, C. D., Raghavan, P. and Schutze, H. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

Schofield, Magnusson, Mimno, 2017 - Pulling Out the Stops: Rethinking Stopword Removal for Topic Models