NLP for political polarity classification from tweets

<latex>\small </latex>

# Introduction and problem understanding

Tweets are a common way for political candidates to express their opinions about current affairs. Since the arrival of the Web 2.0 microblogging platforms have become political instruments and reveal political attitudes of political candidates all over the world (except for those places in which government controls internet access like China). An example of previous work done on the pollical milieu can be found on <latex>\parencite{2010}</latex>, <latex>\parencite{2011}</latex> and <latex>\parencite{22012}</latex>.

A group of Mexican anthropological researchers (Marta Bárbara Ochman et al., 2016) took the task of making a taxonomy just for classifying political attitudes that can be identified on tweets. Their hypothesis is that those attitudes are correlated with future campaign proposals. The taxonomy developed was the following:

1. Proactive: Tweets under this category are aimed to generate information about their personal virtues, their proposed program and their party’s current efforts.
2. Reactive: Seek to neutralize their adversaries’ derisions and any infamous depiction on the media.
3. Aggressive: Emphasize negative traits of their enemies or defame their opponents.
4. Vote winner: Demagogue speech aimed at winning a political advantage.

No one can gainsay that labeling each of these tweets is drudgery. Thus, a model that can classify those tweets with minimal human intervention is highly desirable.

However, this task isn’t easy. Tweets are very limited in the following ways <latex>\parencite{12014}</latex>:

* Data sparsity.
* Changing nature of languages due to trending topics.
* Political candidates usually make use of jargon and informal language.
* Lack of context from the text. The text field is limited to 40 characters.
* Short irregular forms may be used.

Clearly, if we try to use a machine learning classifier over the raw tweets headfirst, the results would be disappointing. So, on the one had we have these data highly skewed and sparse with the problems just mentioned. On the other hand, we have all these mature machine learning algorithms dating back to the 50’s. We need to find good instance representation that our models can use uniformly, expresses latent attributes present in the data and finally reduce noise produced by redundant features that just doesn’t add any value. But how?.

<latex>

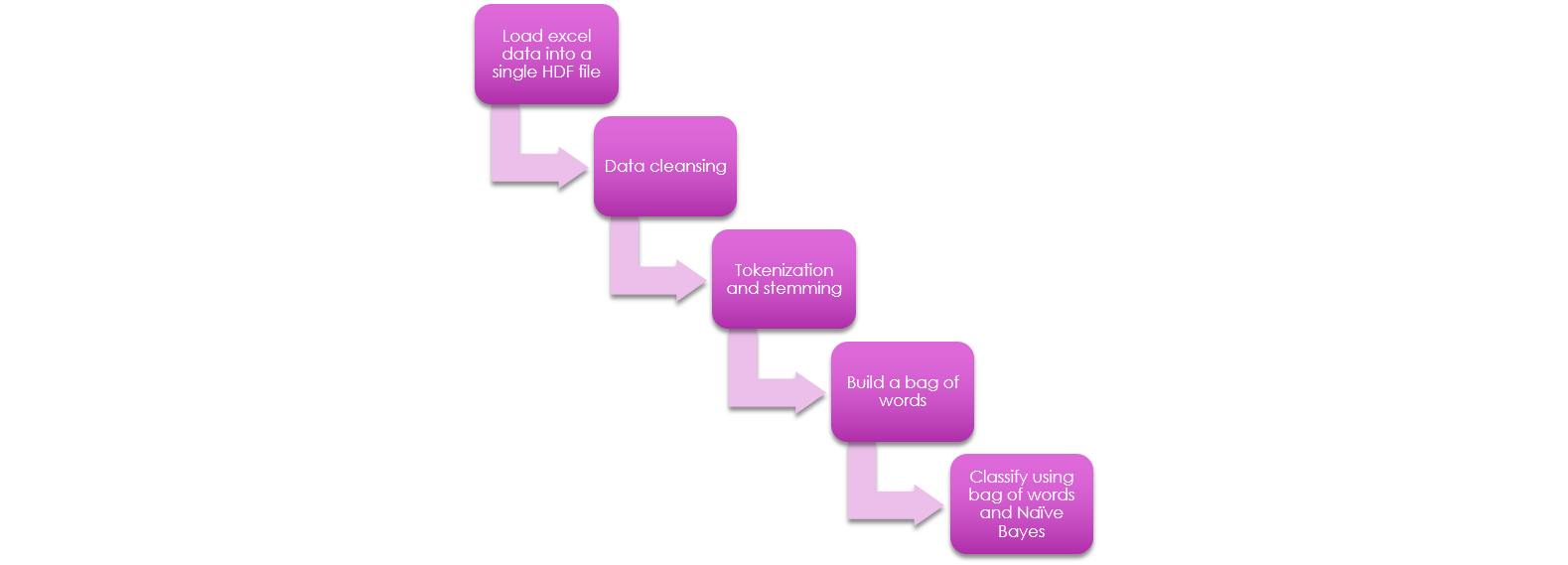
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Text preprocessing pipeline}

\end{figure}

</latex>

Well, firstly we should use some common preprocessing steps that might help semantic and grammar decomposition or somehow incorporate domain knowledge into our search space. These techniques are show in <latex>\Cref{fig:commontasks}</latex>. Secondly, deep learning techniques using CNN have prove to be useful in many realms, from heuristics design to computer vision. NLP is not an exception. Here we can used them to extract those latent features we mentioned earlier. A common technique that has been infallible is word2vec. Since its conception in <latex>\parencite{2013}</latex>, this method has been capable of both preserving semantic and syntactic representation and can be used reliably to categorize a bag-of-words when data is sparse.

The initial hypothesis of the projects is:

1. A set of features represented in a word2vec vector representation of the tweet can leverage the power of an already trained word2vec model and gives a Naïve Bayes classifier a very low generalization error <latexnb>\footnotemark\footnotetext{\nohyph Although there exists an Spanish corpus it is not focused on political jargon and each party has its own jargon}</latexnb>.
2. A more diverse set of features can increase accuracy <latex>\parencite{2018}</latex>. Thus, a minimum representation of a grammatical structure, i.e. a bigram count of *special tokens* are added to the resulting set of features. This bigram count increases the classifiers accuracy.
3. Normalized features achieve better results and can be selected more easily because they are scale invariant. Thus, the vectors corresponding to tweets with different lengths are weighted. However, tweet length will be added to the features in representation of energetic grammatical structures.

Results show that the ROC AUC curve increases significantly, from 0.65 (on average for the four political attitudes) using only bag of words to 0.72 using only bigram features, to 0.85 using word2vec features and finally to 0.857 using all the features.

In the final model we will see that the top 5 most important features are for all the four labels coming from V2W.

# Previous works

<latex>\parencite{1–22008}</latex> places Sentiment Analysis (SA) within the área of Natural Language Processing (NLP) and can be defined as the computational treatment of opinions, feelings, and subjectivity in text. This article mentions that early history places 2001 as the milestone at which a widespread awareness began to arise around sentiment analysis, with beliefs systems as forerunners. One of the factors behind this land rush was the availability of datasets for machine learning on the World Wide Web.

<latex>\parencite{12014}</latex> brings to the table two of the firsts approaches for the research community to tackle the problem of SA. <latex>\parencite{2002b}</latex> proposes the use of linguistic analysis. This kind of approach can be thought as supervised because it relies on previous domain knowledge, e.g. Chomsky grammatical structures. At the other end we have <latex>\parencite{2002a}</latex>, which proposes the use of classical machine learning techniques. Contrary to the approach taken by Turney, here we rely more on achieving a high accuracy using an ensemble of different techniques, commonly ignoring grammatical structures as in the case of a simplification using bag-of-words representation.

The bag-of-words representations gets its name form a passage from linguist Zellig Harris (1954), “language is not merely a bag of words but a tool with particular properties.”. <latex>\parencite{2012}</latex> suggest we think of the model as “putting the words of the training corpus” in a bag and then selecting one word at a time. Then, the notion of order is lost, but we end up with a binary vector that we can neatly use in our machine learning classifiers.

Now, let’s go back to a more recent 5-year horizon. As aforementioned there is an extreme impairment over context in which we are just fettered to a 140 characters text context. Furthermore, tweets usually don’t have representative and syntactically consistent words. <latex>\parencite{102013}</latex> proposes a sentiment grade for each distinct notion in the post using an ontology instead of evaluating it as a whole. The authors use a *Formal Concept Analysis* (FCA) algorithm proposed by <latex>\parencite{1999}</latex> in which applies a user-driven-step-by-step methodology for creating domain models, i.e. it creates an ontology specific for the bulk of tweets to classified. Tweets were classified on a rank per topic. They used a tool called OntoGen in which a semi-supervised approach was possible.

Through the lens of our work, topic and ontologies could prove useful when considering political parties, allies, government institutions, commercial and foreign institutions. However, these ontologies must be built mostly by human annotations, a cost we cannot afford in this study.

The approach taken by <latex>\parencite{42013}</latex> was a bit different. The paper measures how to word of mouth (WOM) affect movies sales, negatively or positively. There were four tweet categories very similar to the ones we are measuring: intention tweets, positive tweets, neutral tweets and negative tweets. *Intention tweets* are very similar to our *vote winner* category because an intention to win votes can be achieved either by aggressive of proactive tweets. The authors decided to use two well-known classical machine learning classifiers: Naïve Bayes and Support Vector Machines. This approach is similar the one proposed by <latex>\parencite{2002a}</latex> in which we harness the efficiency of classical machine learning algorithms by using meaningful instance representations.

In the work of <latex>\parencite{12013}</latex> many approaches for feature extraction are mentioned. Namely, extracting frequent terms while measuring compactness, association rule mining to find syntax rules, ontologies, hyponyms (more general) and meronyms. However, most of the methods mentioned in the introduction use unigrams, ngrams and part-of-speech (POS) <latex>\parencite{12014}</latex>. The next section will explain our approach.

# Proposed algorithm

We propose an ensemble of features that can be better representations than the bag-of-word alone. Although *word2vec* preserves semantic and syntactical relationships, it does not preserve grammatical structures. This drawback can be compensated by just using bigram structures of special tokens in which the order is still maintained. In the following listings we present pseudocode to achieve each piece of the final vector representation that will be used by a Naïve Bayes classifier (vs the classical bag-of-words representation).

<latexnb>

\begin{algorithm}[H].

\caption{ExtractFeatures -Extraction upper process }\label{alg:minerpattern}.

\begin{algorithmic}[1].

\INPUT{$N$ - a tweet, $tokenList$ - list of tokens that should be used}.

\OUTPUT{FS - a set of features}.

\Procedure{ExtractFeatures}{$N$}.

\State $ FS \gets \varnothing $.

\State $ BOW \gets $ ExtractBOW($N.Text$) \Comment{Bag-of-Words extraction}.

\State $ W2V \gets $ ExtractW2V ($N.Text$) \Comment{Word2Vec extraction}.

\State $ BIG \gets $ ExtractBIG ($N.Text$, $tokenList $) \Comment{Bi-gram extraction}.

\State $ FS \gets BOW \cup W2V \cup BIG $ \Comment{Union-all}.

\State $ FS \gets $ Normalize($FS$).

\State\Return FS.

\EndProcedure.

\end{algorithmic}.

\end{algorithm}.

</latexnb>

<latexnb>

\begin{algorithm}[H].

\caption{ExtractBOW -Extract BOW representation}\label{alg:minerpattern}.

\begin{algorithmic}[1].

\INPUT{$N$ - a tweet, $wordList$ - twitter word list}.

\OUTPUT{BOW – a binary vector of words}.

\Procedure{ExtractBOW}{$N$}.

\State $ BOW \gets $ zeros($1$,len($wordList$)).

\State $index \gets 0$.

\Foreach{$i \in wordList$}.

\If{$i \ in N.Text$}.

\State $BOW[index] \gets 1$.

\EndIf.

\State $index++$.

\EndForeach.

\State\Return BOW.

\EndProcedure.

\end{algorithmic}.

\end{algorithm}.

</latexnb>

<latexnb>

\begin{algorithm}[H].

\caption{ExtractW2V -Word2Vec feature extraction }\label{alg:minerpattern}.

\begin{algorithmic}[1].

\INPUT{$N$ - a tweet, $w2v\\_model$ - a neural network pre-trained model that maximizes conditional probability of context given a word}.

\OUTPUT{W2V - a set of features}.

\Procedure{ExtractW2V}{$N$, $w2v\\_model$}.

\State $W2V \gets zeros(1,len(N.Words))$.

\State $index \gets 0$.

\Foreach{$i \in N.Words$}.

\State $W2V[index] \gets $ ApplyW2V($w2v\\_model$, i).

\State $index++$.

\EndForeach.

\State $W2V \gets avg(W2V, axis=1) $ \Comment{Calculate the sentence vector by averaging the vector of their words}.

\State\Return W2V.

\EndProcedure.

\end{algorithmic}.

\end{algorithm}.

</latexnb>

<latexnb>

\begin{algorithm}[H].

\caption{ExtractBIG -Bigram count vector}\label{alg:minerpattern}.

\begin{algorithmic}[1].

\INPUT{$N$ - a tweet, $tokenList$ - list of tokens that should be used}.

\OUTPUT{BIG – bigram counts}.

\Procedure{ExtractBIG}{$N$}.

\State $ bigram\\_list \gets $ GenerateBIG ($tokenList$).

\State $ sentence\\_bigram\\_list \gets $ GenerateBIG ($N.Words$).

\State $BIG \gets zeros(1,len(bigram\\_list))$.

\State $index \gets 0$.

\Foreach{$i \in bigram\\_list $}

\If{$i \in sentence\\_bigram\\_list$}

\State $BIG[index] \gets 1$.

\EndIf

\State $index++$.

\EndForeach

\State\Return BIG.

\EndProcedure.

\end{algorithmic}.

\end{algorithm}.

</latexnb>

<latexnb>

\begin{algorithm}[H].

\caption{ GenerateBIG}\label{alg:minerpattern}.

\begin{algorithmic}[1].

\INPUT{$tokenList$ - list of tokens that should be used}.

\OUTPUT{BIG - a set of features}.

\Procedure{GenerateBIG}{$tokenList$}.

\State global $ specialTokens $ \Comment{Globally defined special part of speech and punctuation}.

\State $ SpecialBigrams \gets $ CombinationsOf2(specialTokens).

\State $ BIG \gets zeros(1,len(SpecialBigrams))$.

\Foreach{$i \in tokenList$}.

\If{not ($i \in specialPOS$)}.

$tokenList.remove(i)$.

\EndIf.

\EndForeach.

\For{$i=0 $ \textbf{to} $ len(tokenList)-1$}.

\For{$j=i+1 $ \textbf{to} $ len(tokenList)$}.

\State $ bigram \gets new $ $ Bigram(tokenList[i], tokenList[j])$.

\State $ pos \gets Find(SpecialBigrams, bigram)$.

\State $ BIG[pos]++ $

\EndFor.

\EndFor.

\State\Return BIG.

\EndProcedure.

\end{algorithmic}.

\end{algorithm}.

</latexnb>

Special tokens to use on the Bigram generator are:

|  |  |
| --- | --- |
| Token | Purported Attitude |
| ellipsis | Reactive |
| exclamation | Aggressive, Proactive |
| hashtag | Proactive, Vote Winner |
| mention | Proactive, Reactive |
| name | Aggressive, Vote Winner |
| neg\_emoticon | Aggressive |
| pos\_emoticon | Proactive, Vote Winner |
| question | Proactive |
| quoted | Vote Winner |
| uppercase | Aggressive |
| url | Proactive |
| colon | Vote Winner, Proactive |
| semicolon | Aggressive |
| comma | Vote Winner |

These tokens will also be counted individually (1-gram). In contrast to other approaches found on the internet <latex>\parencite{2017b}</latex>, the bigrams will be counted instead of just asserting their presence.

We can see that the algorithms are quite simple. This owns to the fact that we are relying more on the pretrained neural networks models that came with the gensim python library for word2vec representations. Those packages already came pretrained on <latex>\parencite{2017c}</latex>.

# Experimental setup

Our dataset is small. From over 51,453 samples extracted from the provided Excel files, we just have labels for only 7,594 of them. Due to the skewness we have decided to make a stratified sample set consisting of <latex>$\frac{3}{5}$</latex> of the data for training and <latex>$\frac{2}{5}$</latex> thirds for testing, i.e. 4,500 and 3,094 respectively.

Ground truth consists of a rank given for each of the four categories in each tweet. These ranks go from “0” to “9” and can be considered as ordinal values. As far as this study is concerned, no correlation exists between these 4 target classes. Thus, we will treat each category as a separate classification problem.

Some tweets present an homogeneous structure (having only one class dominate over the others) while other tweets are more ambiguous. The following figure shows the 4 target classes distribution in a binary way, “0” equals “not present” and “not 0” equals “present”:

<latex>

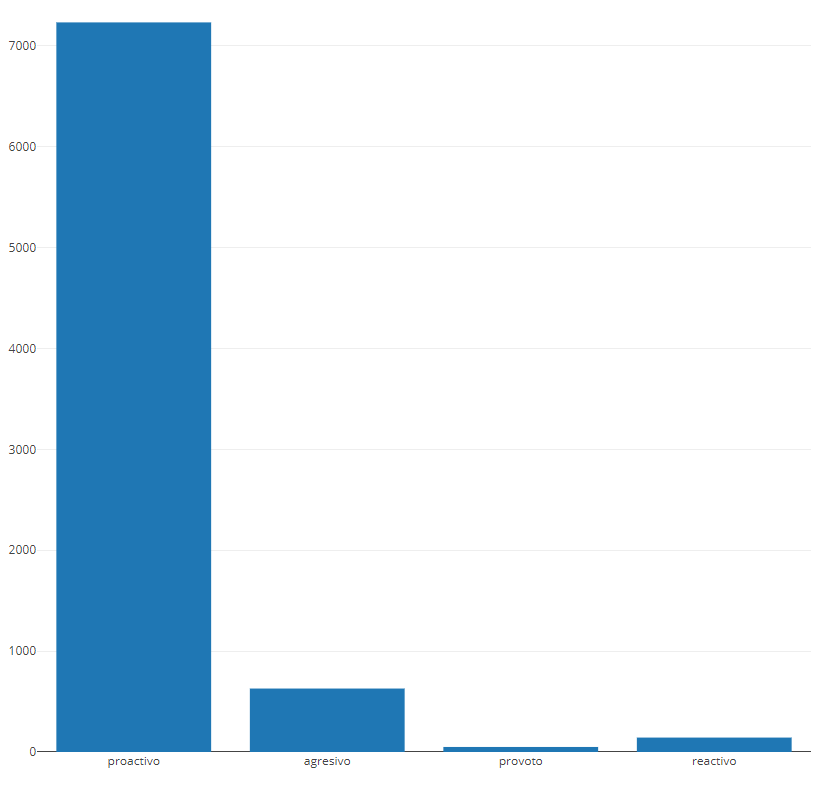
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Data is heavily skewed towards proactive attitude. The approach given in \cref{proposed-algorithm} would help us ameliorate this problem.}

\end{figure}

</latex>

In most of the provided files there were three human annotators. However, through a thorough database perusal it was found that only Martha had valid annotations. So, let’s do a variance analysis over each of the target classes at least for Martha.

<latex>

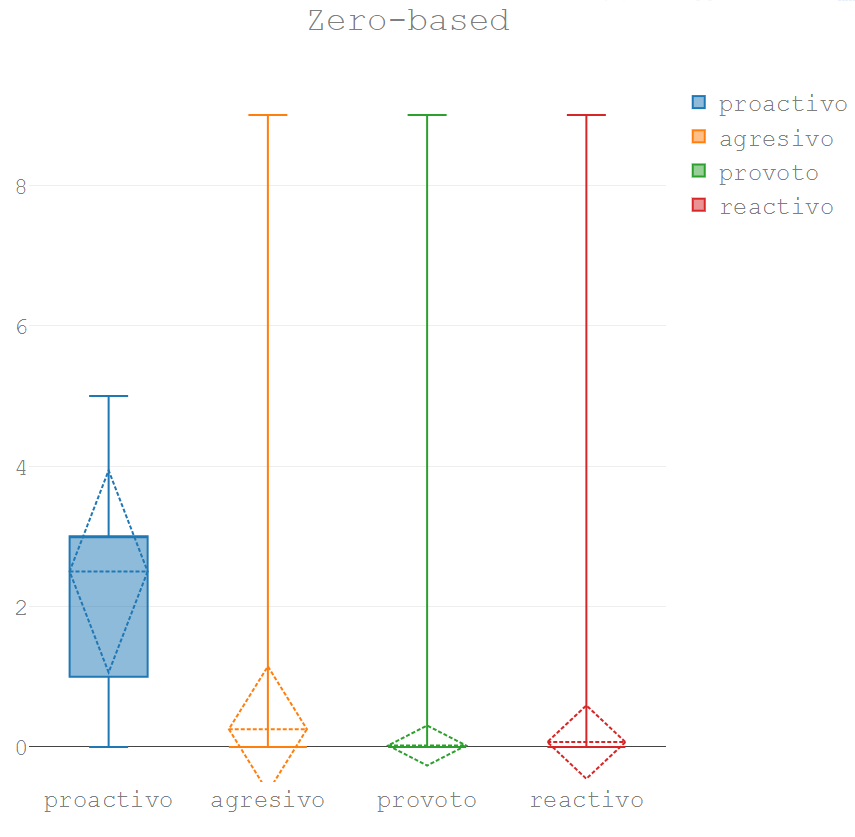
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Overall category rankings provided by Martha}

\end{figure}

</latex>

Just as expected, Martha is mostly filling the proactive field without further care for any of the other categories. The rest of the categories have a mean close to zero. However, a comparable variance analysis can be done filtering out the times when those categories weren’t used and taking into account only the tweets which have a value greater than zero.

<latex>

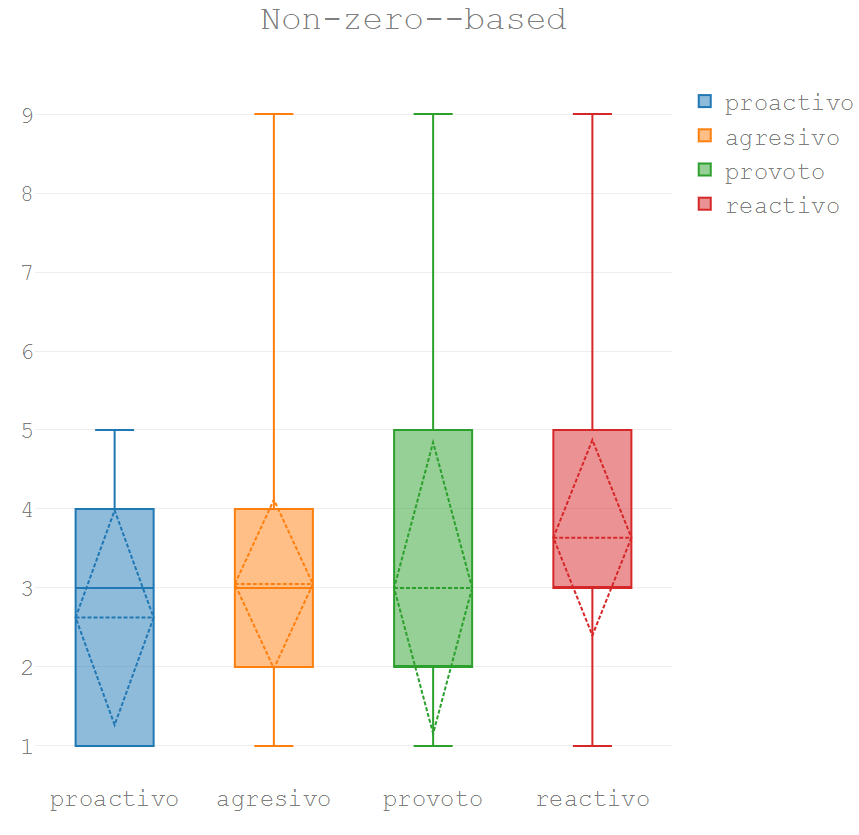
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{W hen Martha decides to use some category ($rank \neq 0$), which numbers she uses more frequently}

\end{figure}

</latex>

Here are some observations we can derive from the graph:

* While the most used category is proactive, Martha is still quite reserved in the scores she gave on this category, reaching a peak at 5.
* The vote-winner category has a similar interquartile range but reaches a more radical peak at 9.
* The reactive and aggressive categories swing wildly but their means are between 3 and 4.

Now let’s explore some the most common words that can serve of basis for a bag-of-words representation. We will remove the stop words because they are too common (low inverse document frequency) and do not add value to the contrast we are trying to discover.

<latex>

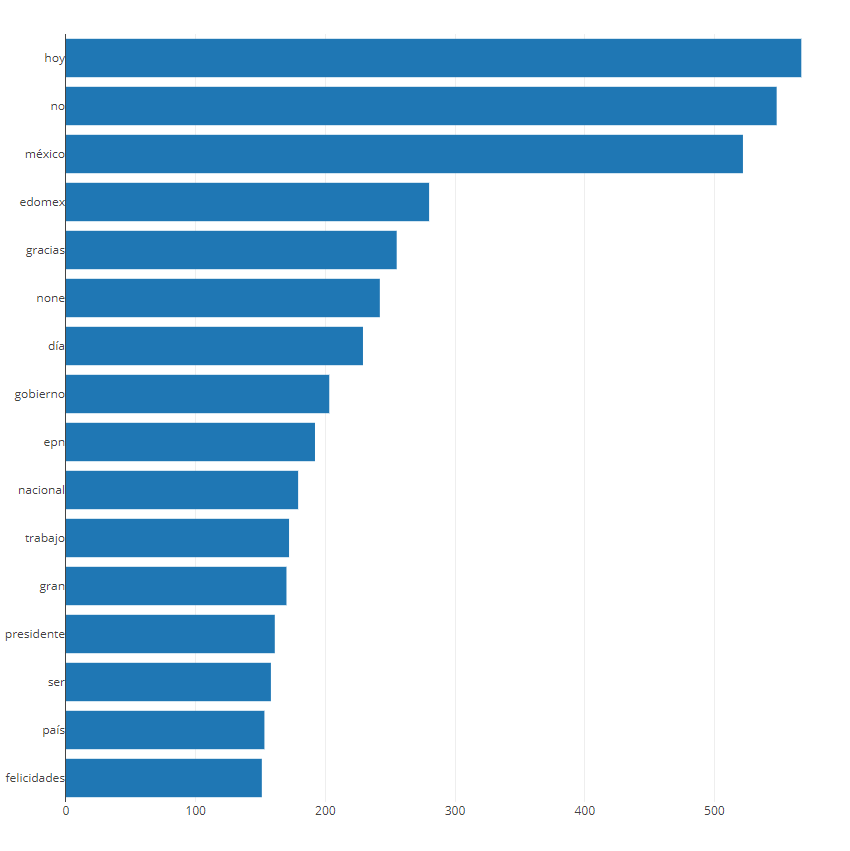
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Bar chart showing the most frequent used word}

\end{figure}

</latex>

However, this word list still contains political parties and names. Let´s remove them and split by category usage.

<latex>

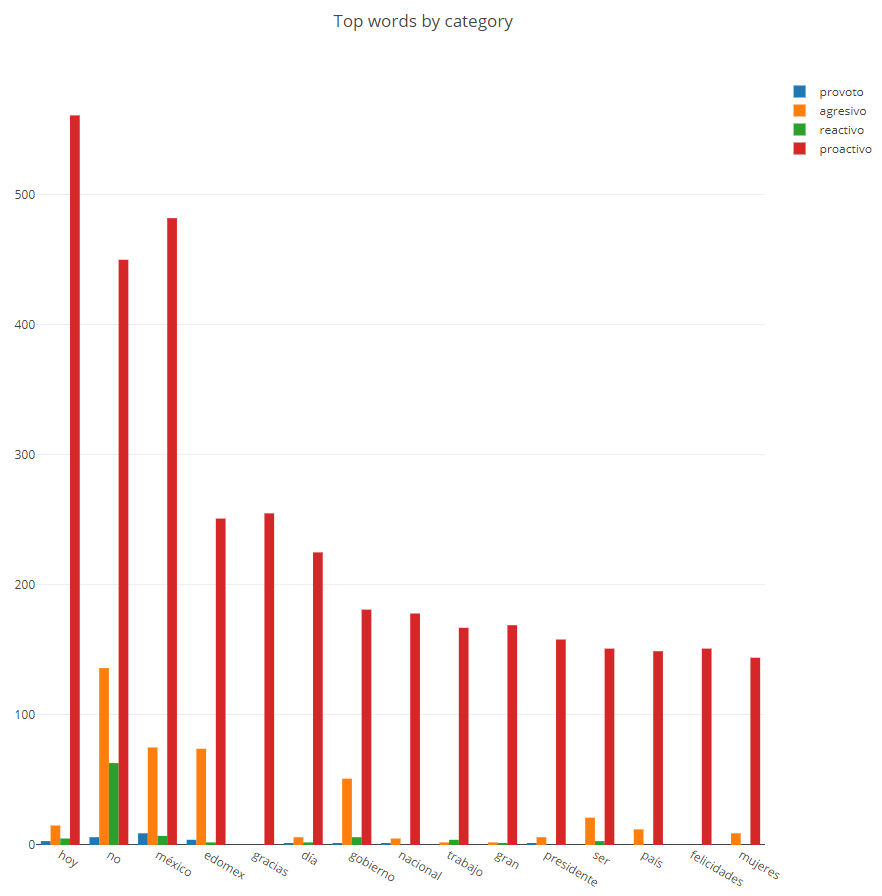
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Most used words per category}

\end{figure}

</latex>

<latex>

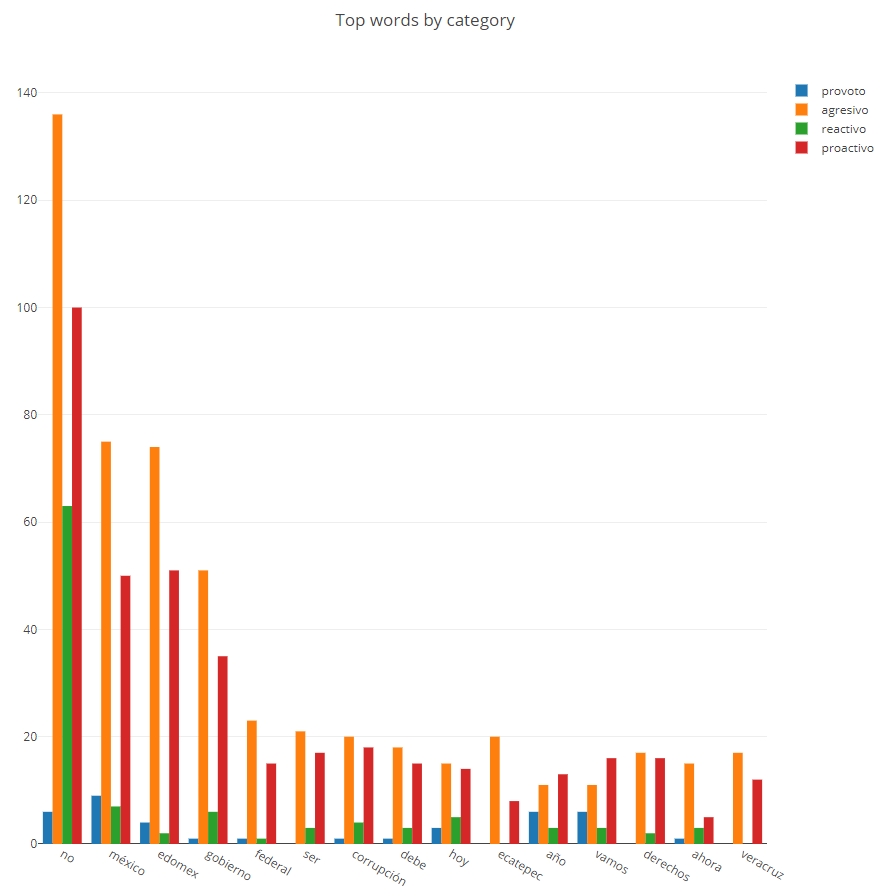
\begin{figure}[H]

\label{fig:commontasks}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Word usage per category taking into account that at least that word was used in other category from proactive }

\end{figure}

</latex>

Owning to the fact that we added the word “no” to the exception list is the most common word.

A ROC curve will be used to measure accuracy selecting different set of features, once using a Naïve Bayes classifier and once using Random Forest. Naïve Bayes classifiers works well when using prior binary knowledge of which words are present in a sentence, even using a bag-of-words and is probable it will get more uniform results. To complete our test Random Forest will exploit the features more uniformly and probably will get a significant improvement.

It is impossible to measure the statistical significance of the improvement because we are only using one database. According to <latex>\parencite{12011}</latex> we need at least <latex>$n\eq a\* k$</latex> where <latex>$a$</latex> is a number between 2 and 8 and <latex>$k$</latex> is the number of classifiers we are testing. Wilcoxon test, which deals for just two classifiers as in our case, also relies on this number of separate databases.

Finally, we are going to measure feature importance in the accuracy of each classifier.

# Results and discussion

From a source code view (PyCharm project provided), the process taken to generate the results where:

1. In the *cleansing* python package run the files in the following order:
   1. **Load\_tweets.py**. Centralizes the information contained in the provided Excels files into a one hdf5 and then a pickle file called final.pickle.
   2. **Tockenize\_tweets.py**. Generates the column *tokenized\_text* which is the text array in which has cleansed text for the BOW extraction.
   3. **Generate\_tokenized\_text\_noparties.py**. Removes stopwords and political party names from the *tokenized\_text* column.
   4. **Categories\_boxplot.py**. Visualize Martha preference treating the classes as if they were numerical.
2. The *testing* package has some files we can use at this point. Those are:
   1. **Wordlist\_histogram.py**. Plot that shows the most frequent words that will be pprobably important for our classifier.
   2. **Wordlist\_category\_histogram.py**. Help us visualize the problem in terms of most frequent words distribution over the 4 attitudes.
   3. **Box\_plots.py**. Visualize Martha preference contrasting attribute usage by the presence of zero.
3. Here comes our contribution. In the *features* packages there are the following files:
   1. **ExtractBOW.py**. Binary bag-of-words vector extraction. All the extracted features are saved in Excel and pickle for latter usage.
   2. **SpecialTockens.py**. This file contain the function that we are going to evaluate in order to compute the special tokens ngram model.
   3. **ExtractBIG.py**. Common sense features are extracted first by counting special tokens. Then bigrams are extracted and counted to preserve grammatical structures that are obviated by the W2V representation.
   4. **ExtractW2V.py**. Spanish word2vec vector representation are extracted for each word and then and an average overall tweet vector is calculated.
   5. **ExtractFeatures.py**. Consolidate all the features into one file.
   6. **Save.py**. Save the features to an Excel and a Pickle file for rapid usage.
4. Having generated all the features, lets go back to our *testing* package:
   1. **Test\_classifier.py** . Contains base functions for calculating the *ROC AUC* for multiclass environment, as well as **precision**, **recall**, **accuracy** and **f1-scores** that can be appreciated indirectly on the confusion matrices. For the sake of conciseness, the only metric presented on the report was the *ROC AUC*. A cross validation function is provided.
   2. **Confusion\_matrix.py**. Two graph type were generated, the general one presented on <latex>\Cref{generalresults}</latex> and the confusion matrices that will be explained in due course.
   3. **Feature\_importance.py**. Finally this file help us visualize what are the most relevant top 20 features for the *W2V+BIG+BOW RandomForest*.

Having explored the data we saw there is a tendency for proactive classification. Thus, we should expect having low recall values for the aggressive, vote-winner and reactive classes. Two classifiers will be tested, a Naïve Bayes classifier that adapts very well to binary features and a Random-Forest one that can use features more uniformly. Both are natively multiclass. Random Forest was trained with 100 trees.

In this section we evaluate each of the feature sets individually taking into account also the class there are trying to predict.

<latex>

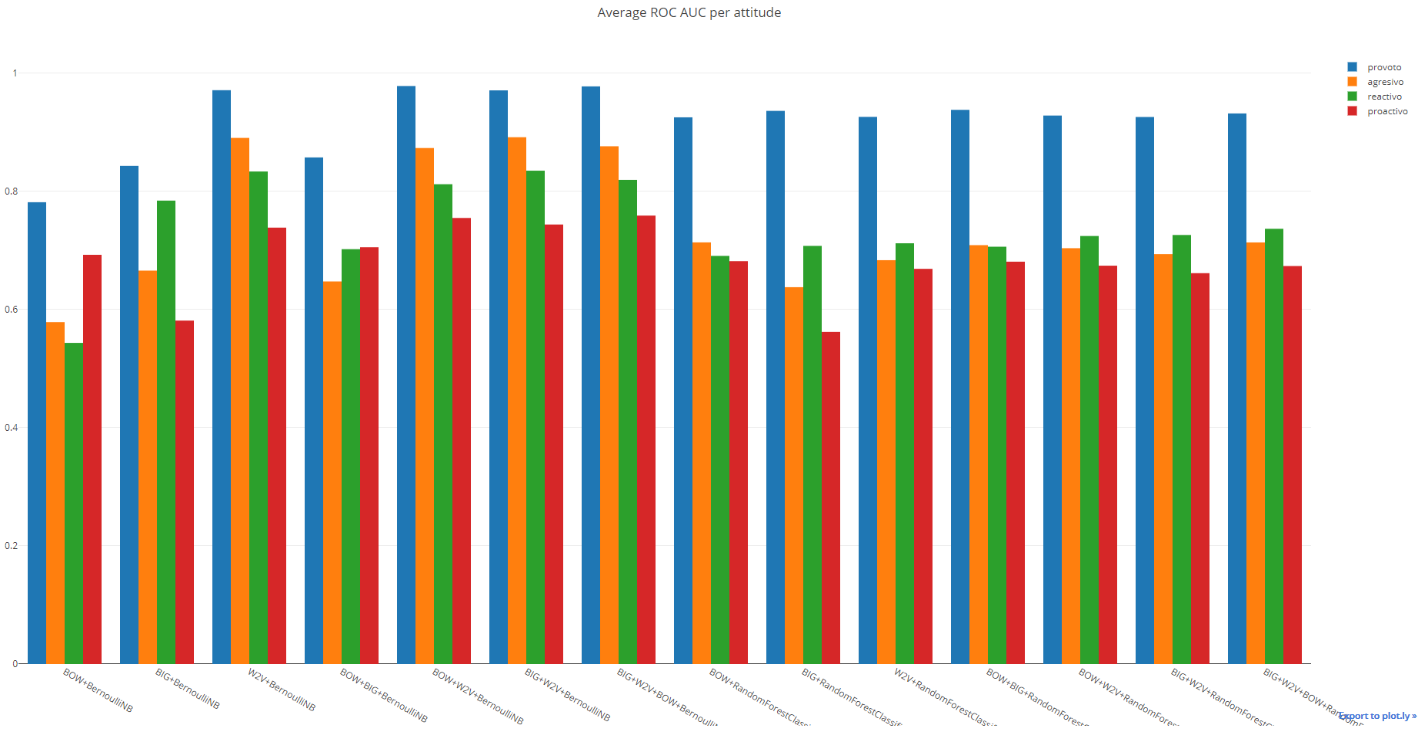
\begin{figure}[H]

\label{generalresults}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

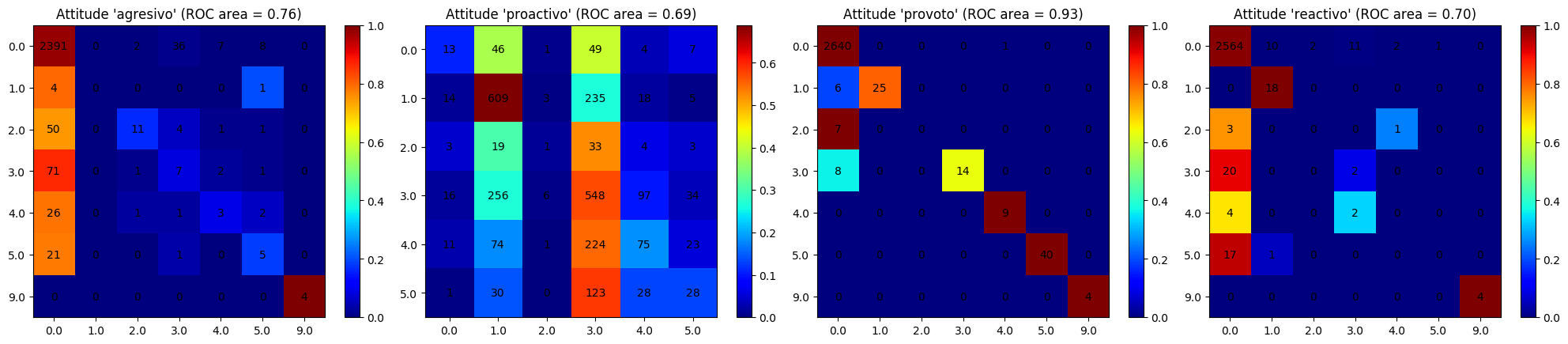
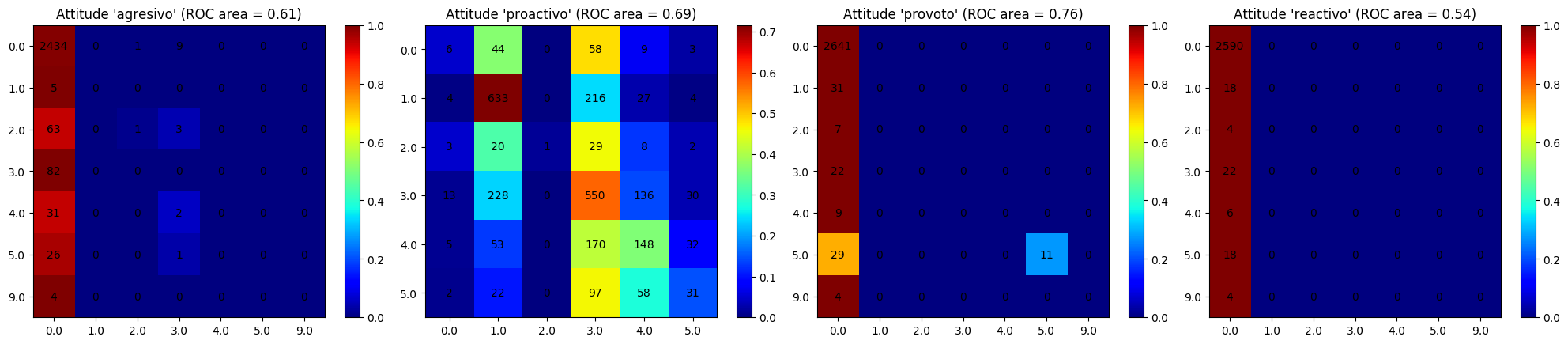
\caption{Results obtained expressed as the macro average ROC area using a one vs rest classifier (Weka approach)}

\end{figure}

</latex>

General ROC areas (one vs rest) demonstrate that BIG+W2V using Naïve Bayes has a slightly better accuracy than the rest of the features. BOW features actually worsened the classifier accuracy and so do using Random Forest. However, literature recommends using confusion matrices for measuring performance on multiclass classifiers. Let’s derive some conclusions from looking them.

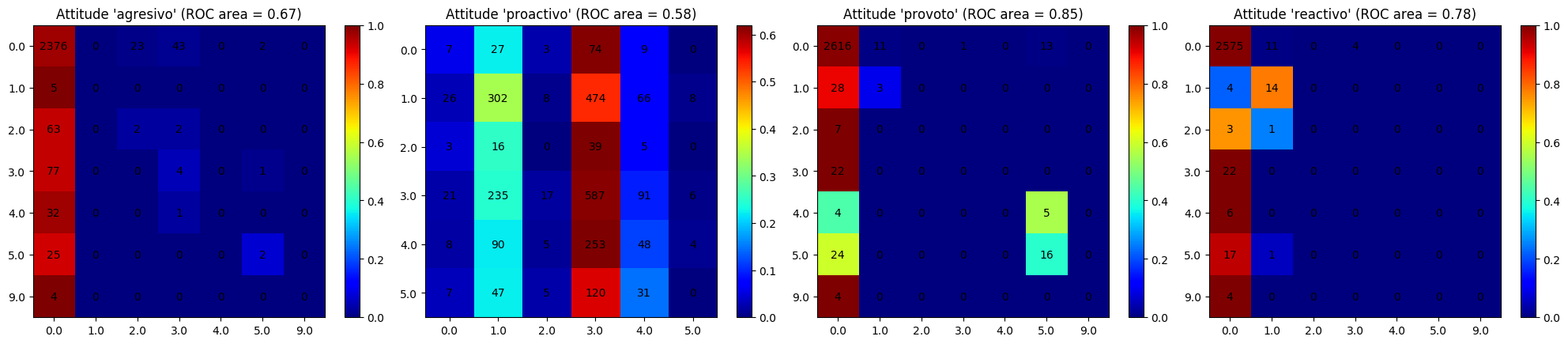
## Bag-of-Words

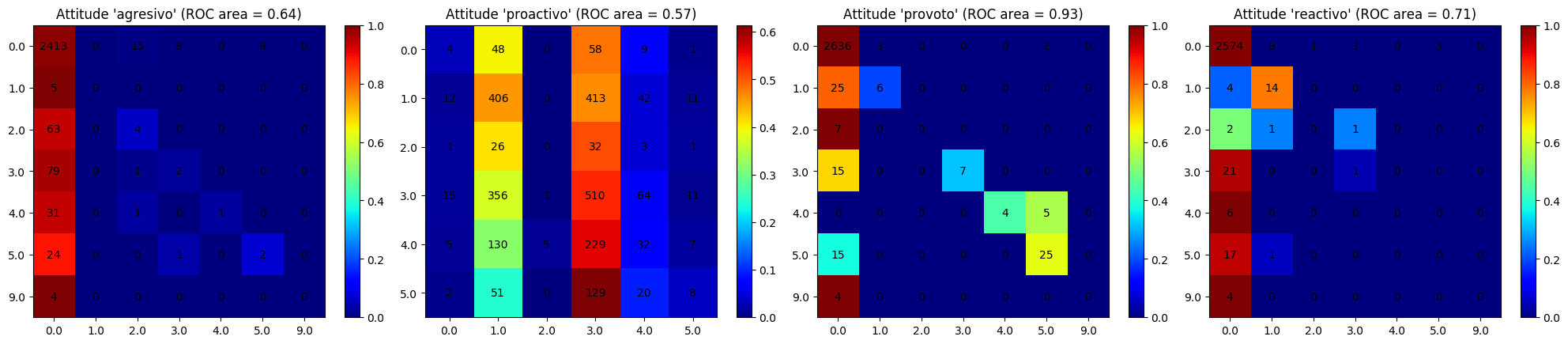


The first row of the confusion matrix heatmap show the BernoulliNB classifier and the second one the RandomForestClassifier.

It is no surprise that bag-of-word has an good performance on the *Proactive* label. However, the actual data has a penchant for 3 so our classifier fails to pickup this static tendency. *Vote winner* category improves significantly when we use maybe owning to the fact that RandomForest uses the features better.

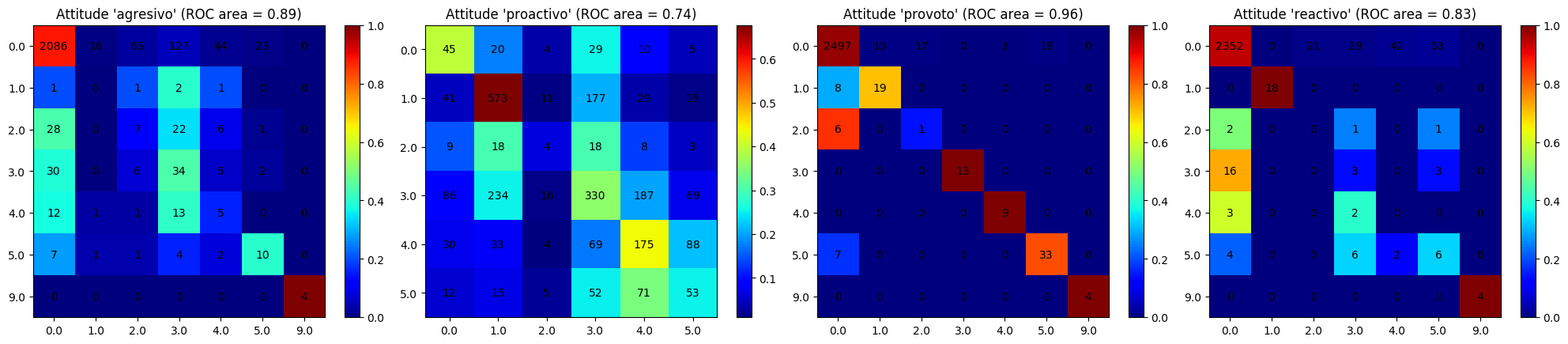
## Bigrams

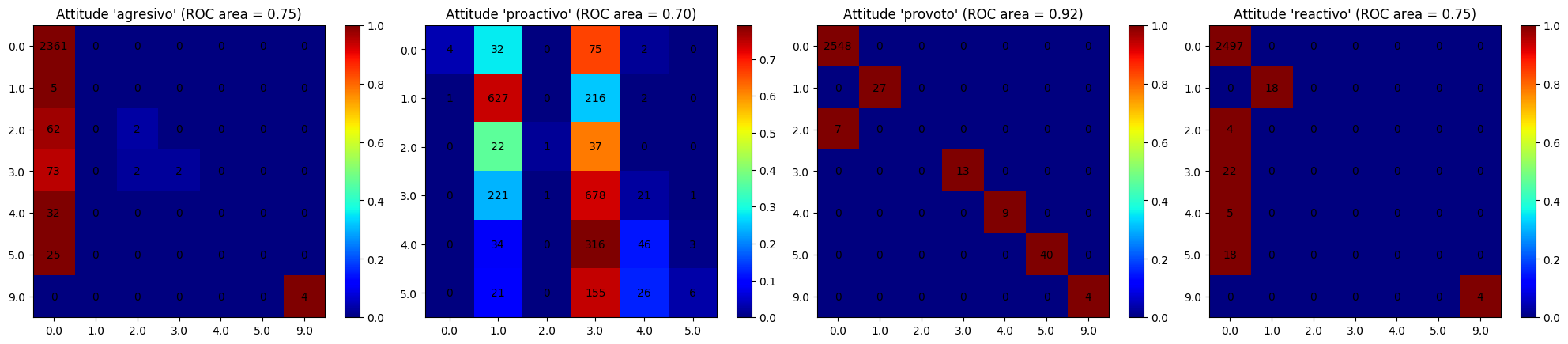




Compared to BOW, Bigrams improve the aggressive and the provote and reactive categories using *Naïve Bayes*. However, using *Random Forest* the behavior is a little worse.

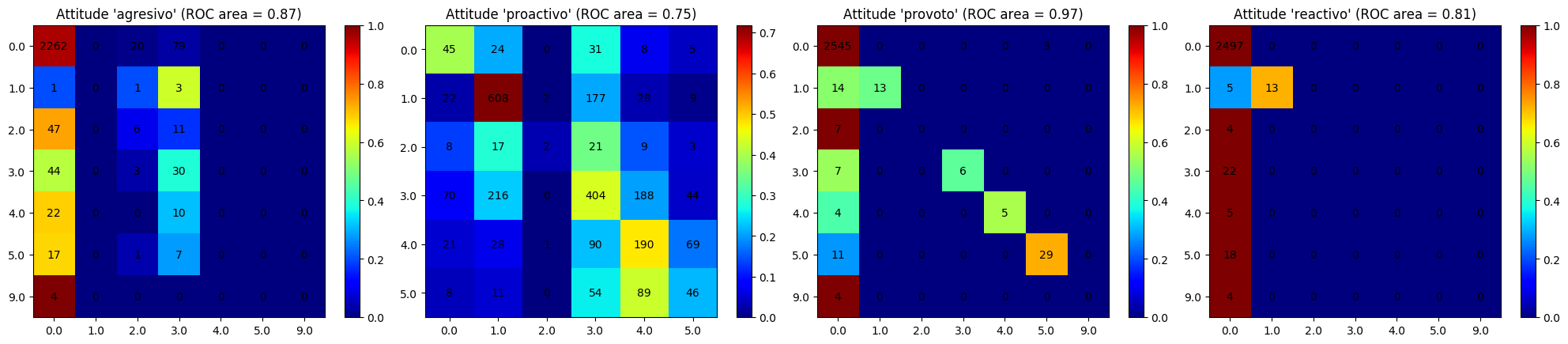
## Word2Vec

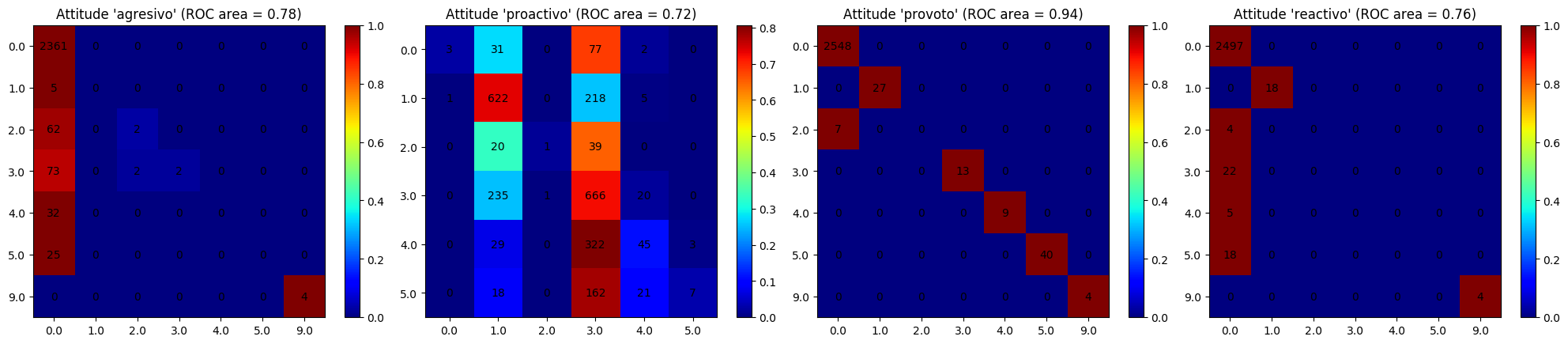




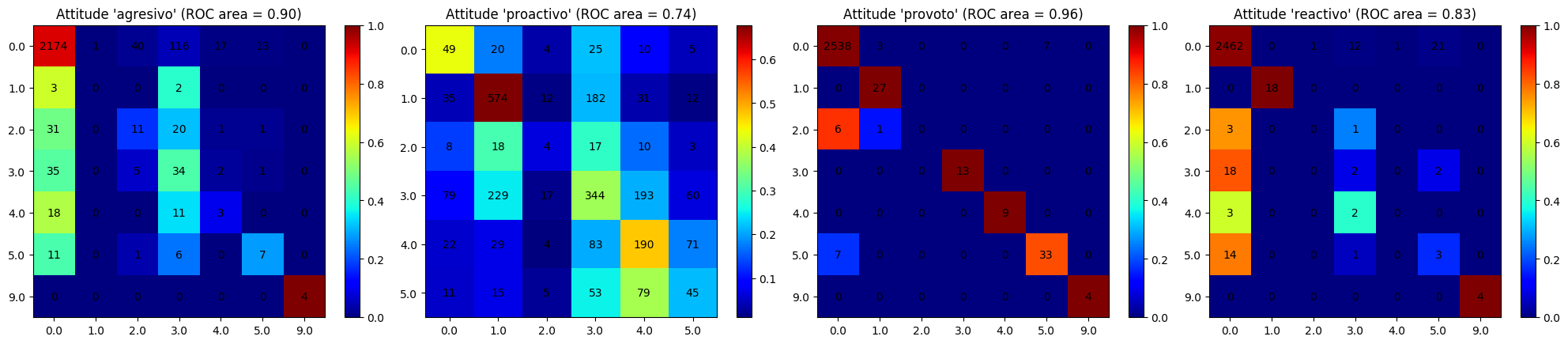
*Word2vec* features improve substantially both NB and RF. *Provoto* precision and recall levels also seem to improve.

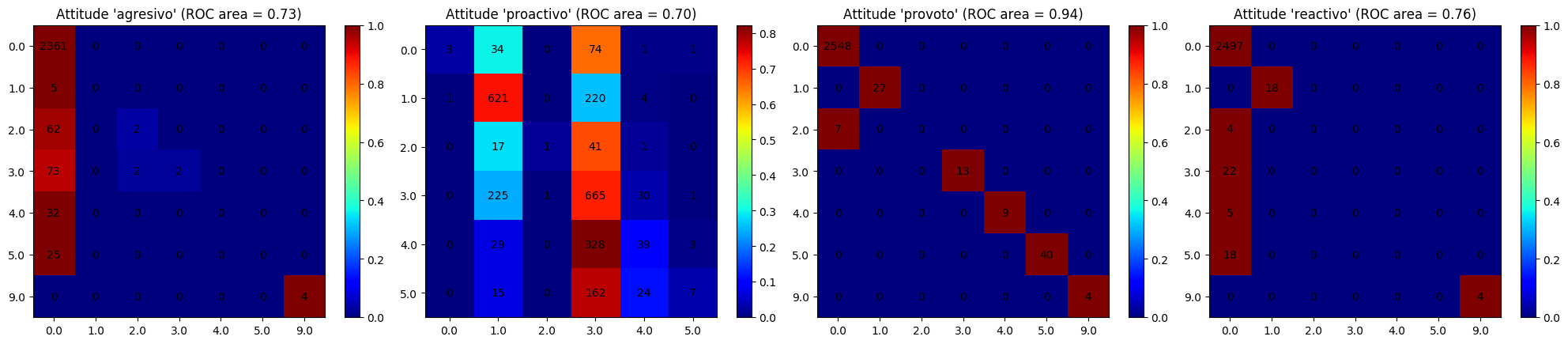
## Word2Vec + BOW



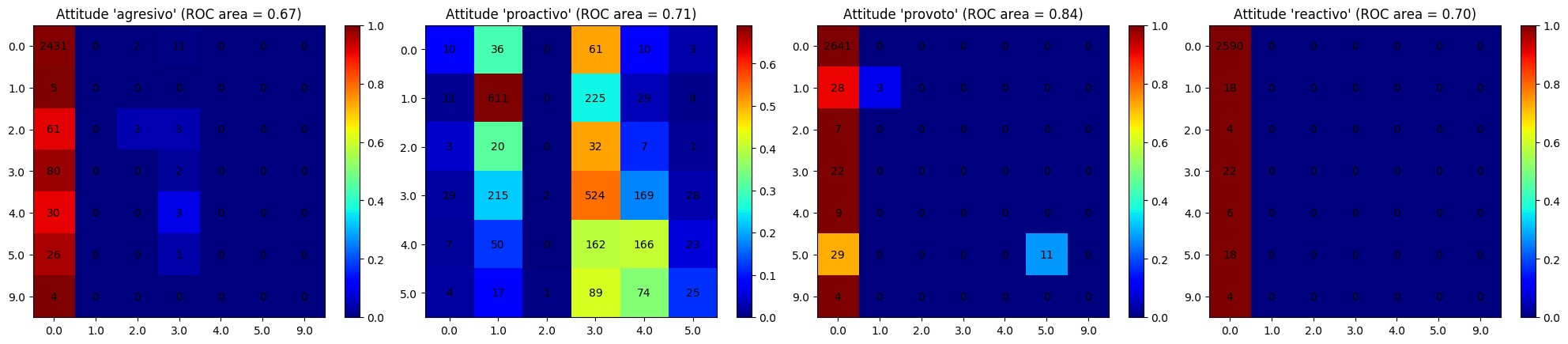


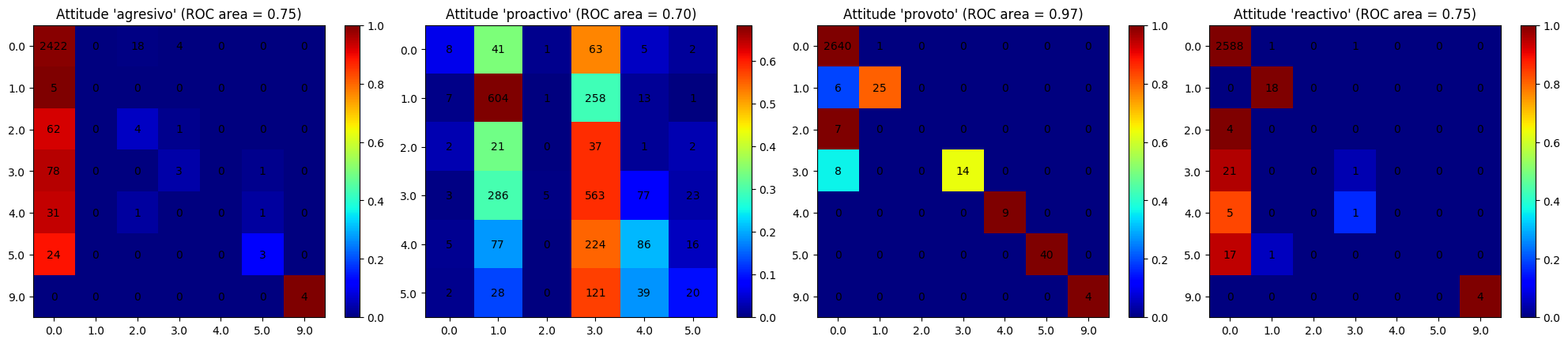
## Word2Vec + Bigrams



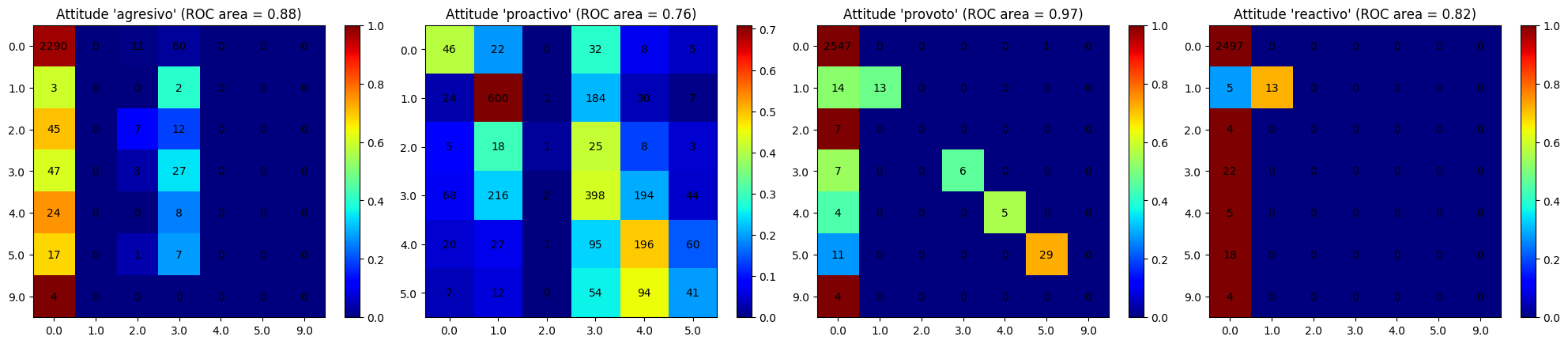


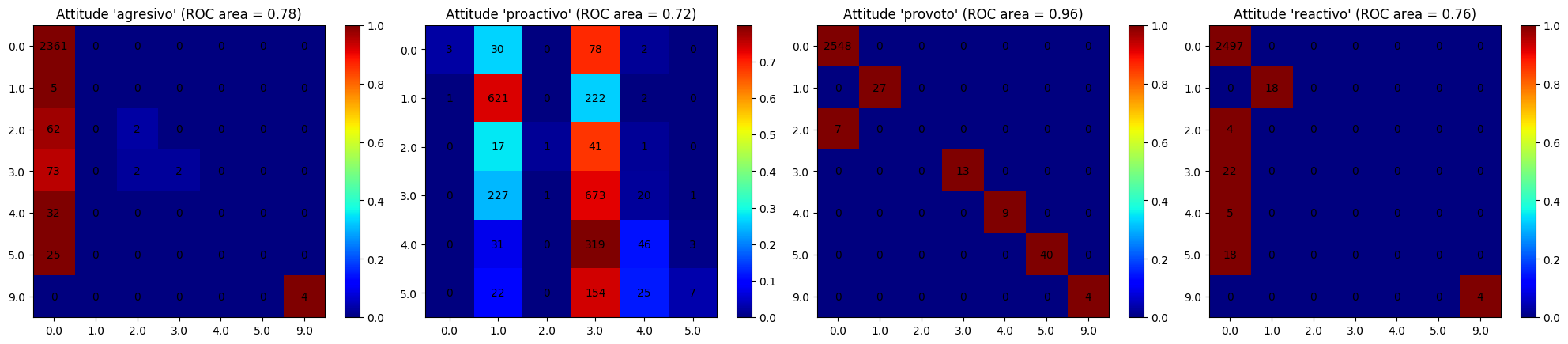
## BOW + Bigrams





Mixed features (BOW + W2V + BIG)





All the features combine generate a nearly perfect *Provoto* classification for *RandomForest*. However, that is as far as it gets, because the data is insufficient for learning other classes. The vertical lines indicate that the actual class is always static, the source of a big generalization error that could be ameliorated if we had more data.

# Feature importance

We can see from the following figures that that most helpful features come from the word2vec classifier. Some come from BIG and a bit fewer from BOW.

<latex>

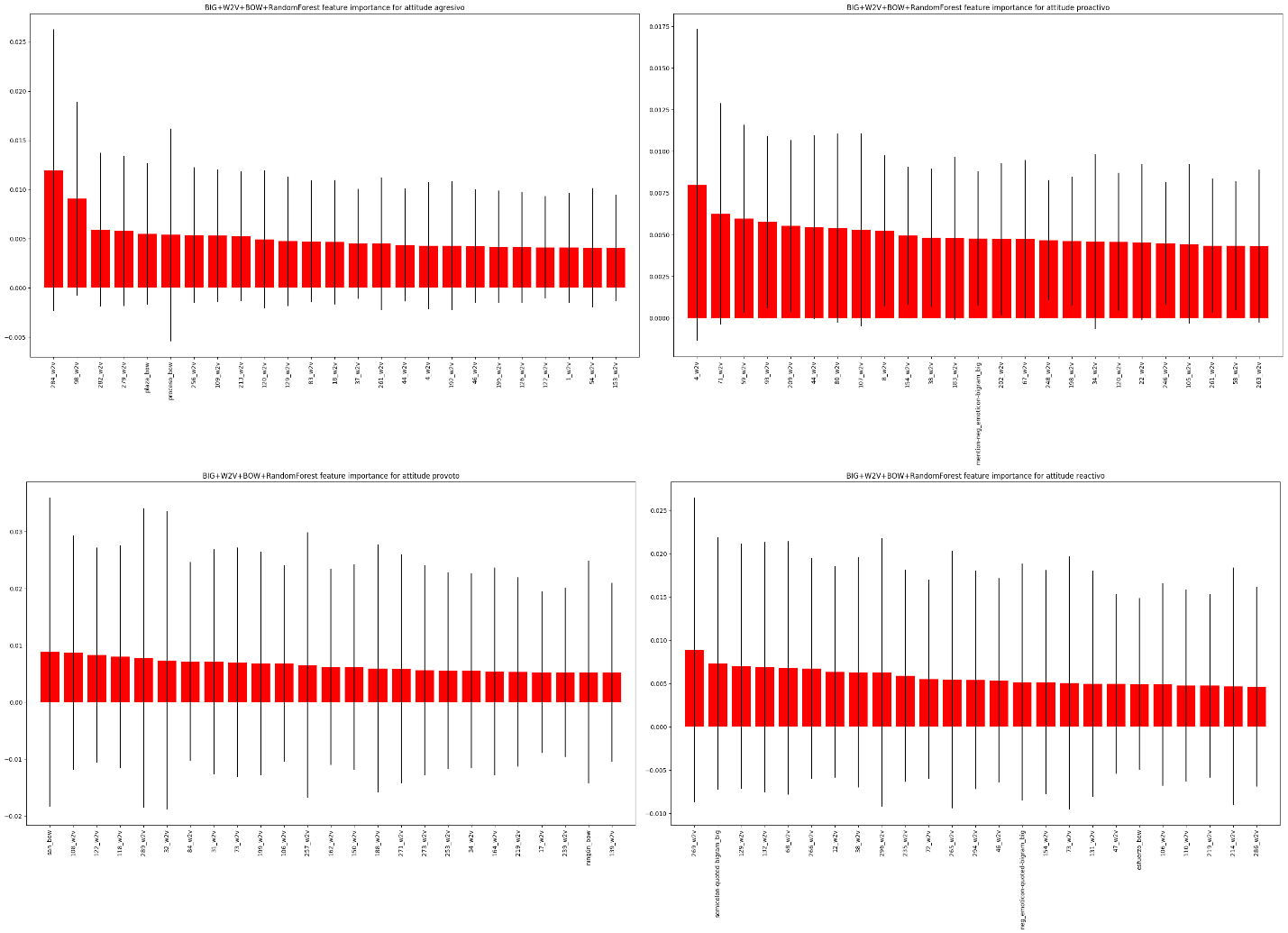
\begin{figure}[H]

\label{generalresults}

\centering

\begin{minipage}{\textwidth}

</latex>



<latex>

\end{minipage}

\caption{Feature importance}

\end{figure}

</latex>

# Conclusions

Features with which we feed a machine learning algorithm are very important. We just saw how by just adding bigram features improved accuracy and *ROC AUC* consistently. Feature work may generate more levels of bigrams and then made feature extraction by importance to improve the *recall* of the classifier.

Word2vec vector representation can help integrate the syntactic and semantic structures of any language and find similarities between sentences. Those similarities can then be used to train a more complex classifier.

The vertical lines in our results indicate that the actual class is static and that there is a big generalization error, so we probably we could do better with more data.

# Reference work listing

An interesting study on political analysis of tweets was done by <latex>\parencite{2017a}</latex> in which a Spanish tweets were used in conjunction with word2vec for political elections prediction. They measure their results over the F1-score and using Support Vector Machines. On two distinct phases of 2242 and 16 million tweets, they achieved 60.5% and 85.09% respectively. A similar approach of building n-grams was used. Compared to our results they achieved better results but that can be explained by the nature of their problem which was a single class problem.

Another starting point was the work published in <latex>\parencite{2017b}</latex>. However, in that work they don’t use bigram generation nor bigram counting to feed the classifiers.

# References

<bibliography> @article{2011, title={On Using Twitter to Monitor Political Sentiment and Predict Election Results}, abstractNote={The body of content available on Twit- ter undoubtedly contains a diverse range of political insight and commentary. But, to what extent is this representative of an electorate? Can we model political sentiment effectively enough to capture the voting intentions of a nation during an election capaign? We use the recent Irish General Election as a case study for investigating the potential to model political sentiment through mining of social media. Our approach combines sentiment analysis using supervised learning and volume-based measures. We evaluate against the conventional election polls and the final election result. We find that social analytics using both volume-based measures and sentiment analysis are predictive and we make a number of observations related to the task of monitoring public sentiment during an election campaign, including examining a variety of sample sizes, time periods as well as methods for qualitatively exploring the underlying content.}, journal={Psychology}, author={Bermingham, Adam and Smeaton, Alan F}, year={2011}, pages={2–10}}

@article{12011, title={A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms}, volume={1}, ISBN={2210-6502}, ISSN={22106502}, url={http://dx.doi.org/10.1016/j.swevo.2011.02.002}, DOI={10.1016/j.swevo.2011.02.002}, abstractNote={The interest in nonparametric statistical analysis has grown recently in the field of computational intelligence. In many experimental studies, the lack of the required properties for a proper application of parametric procedures - independence, normality, and homoscedasticity - yields to nonparametric ones the task of performing a rigorous comparison among algorithms. In this paper, we will discuss the basics and give a survey of a complete set of nonparametric procedures developed to perform both pairwise and multiple comparisons, for multi-problem analysis. The test problems of the CEC’2005 special session on real parameter optimization will help to illustrate the use of the tests throughout this tutorial, analyzing the results of a set of well-known evolutionary and swarm intelligence algorithms. This tutorial is concluded with a compilation of considerations and recommendations, which will guide practitioners when using these tests to contrast their experimental results. © 2011 Elsevier B.V. All rights reserved.}, number={1}, journal={Swarm and Evolutionary Computation}, publisher={Elsevier B.V.}, author={Derrac, Joaquín and García, Salvador and Molina, Daniel and Herrera, Francisco}, year={2011}, pages={3–18}}

@book{1999, title={Formal concept analysis : mathematical foundations}, ISBN={3540627715}, url={https://dl.acm.org/citation.cfm?id=550737}, abstractNote={This is the first textbook on formal concept analysis. It gives a systematic presentation of the mathematical foundations and their relation to applications in computer science, especially in data analysis and knowledge processing. Above all, it presents graphical methods for representing conceptual systems that have proved themselves in communicating knowledge. Theory and graphical representation are thus closely coupled together. The mathematical foundations are treated thoroughly and illuminated by means of numerous examples. 0. Order-theoretic Foundations -- 1. Concept Lattices of Contexts -- 2. Determination and Representation -- 3. Parts and Factors -- 4. Decompositions of Concept Lattices -- 5. Constructions of Concept Lattices -- 6. Properties of Concept Lattices -- 7. Context Comparison and Conceptual Measurability.}, publisher={Springer}, author={Ganter, Bernhard. and Wille, Rudolf.}, year={1999}}

@article{22012, title={Why the Pirate Party Won the German Election of 2009 or The Trouble With Predictions: A Response to Tumasjan, A., Sprenger, T. O., Sander, P. G., &amp; Welpe, I. M. “Predicting Elections With Twitter: What 140 Characters Reveal About Political Sentiment”}, volume={30}, ISBN={9780769545783}, ISSN={0894-4393}, url={http://journals.sagepub.com/doi/10.1177/0894439311404119}, DOI={10.1177/0894439311404119}, number={2}, journal={Social Science Computer Review}, author={Jungherr, Andreas and Jürgens, Pascal and Schoen, Harald}, year={2012}, pages={229–234}}

@article{2017a, title={Classification of Spanish election tweets (COSET) 2017: Classifying tweets using character and word level features}, volume={1881}, ISSN={16130073}, journal={CEUR Workshop Proceedings}, author={Khandelwal, Ankush and Swami, Sahil and Akhtar, Syed S. and Shrivastava, M.}, year={2017}, pages={49–54}}

@article{102013, title={Ontology-based sentiment analysis of twitter posts}, volume={40}, ISBN={09574174}, ISSN={09574174}, url={http://dx.doi.org/10.1016/j.eswa.2013.01.001}, DOI={10.1016/j.eswa.2013.01.001}, abstractNote={The emergence of Web 2.0 has drastically altered the way users perceive the Internet, by improving information sharing, collaboration and interoperability. Micro-blogging is one of the most popular Web 2.0 applications and related services, like Twitter, have evolved into a practical means for sharing opinions on almost all aspects of everyday life. Consequently, micro-blogging web sites have since become rich data sources for opinion mining and sentiment analysis. Towards this direction, text-based sentiment classifiers often prove inefficient, since tweets typically do not consist of representative and syntactically consistent words, due to the imposed character limit. This paper proposes the deployment of original ontology-based techniques towards a more efficient sentiment analysis of Twitter posts. The novelty of the proposed approach is that posts are not simply characterized by a sentiment score, as is the case with machine learning-based classifiers, but instead receive a sentiment grade for each distinct notion in the post. Overall, our proposed architecture results in a more detailed analysis of post opinions regarding a specific topic. © 2012 Elsevier Ltd. All rights reserved.}, number={10}, journal={Expert Systems with Applications}, author={Kontopoulos, Efstratios and Berberidis, Christos and Dergiades, Theologos and Bassiliades, Nick}, year={2013}, pages={4065–4074}}

@article{12013, title={Deriving market intelligence from microblogs}, volume={55}, ISBN={0167-9236}, ISSN={01679236}, url={http://dx.doi.org/10.1016/j.dss.2013.01.023}, DOI={10.1016/j.dss.2013.01.023}, abstractNote={Given their rapidly growing popularity, microblogs have become great sources of consumer opinions. However, in the face of unique properties and the massive volume of posts on microblogs, this paper proposes a framework that provides a compact numeric summarization of opinions on such platforms. The proposed framework is designed to cope with the following tasks: trendy topics detection, opinion classification, credibility assessment, and numeric summarization. An experiment is carried out on Twitter, the largest microblog website, to prove the effectiveness of the proposed framework. We find that the consideration of user credibility and opinion subjectivity is essential for aggregating microblog opinions. The proposed mechanism can effectively discover market intelligence (MI) for supporting decision-makers. © 2013 Elsevier B.V. All rights reserved.}, number={1}, journal={Decision Support Systems}, publisher={Elsevier B.V.}, author={Li, Yung Ming and Li, Tsung Ying}, year={2013}, pages={206–217}}

@article{12014, title={Sentiment analysis in Twitter}, volume={20}, ISBN={1351-3249r1469-8110}, ISSN={1351-3249}, url={http://www.journals.cambridge.org/abstract\_S1351324912000332}, DOI={10.1017/S1351324912000332}, abstractNote={In recent years, the interest among the research community in sentiment analysis (SA) has grown exponentially. It is only necessary to see the number of scientific publications and forums or related conferences to understand that this is a field with great prospects for the future. On the other hand, the Twitter boom has boosted investigation in this area due fundamentally to its potential applications in areas such as business or government intelligence, recommender systems, graphical interfaces and virtual assistance. However, to fully understand this issue, a profound revision of the state of the art is first necessary. It is for this reason that this paper aims to represent a starting point for those investigations concerned with the latest references to Twitter in SA.}, number={1}, journal={Natural Language Engineering}, author={MARTÍNEZ-CÁMARA, EUGENIO and MARTÍN-VALDIVIA, M. TERESA and UREÑA-LÓPEZ, L. ALFONSO and MONTEJO-RÁEZ, A RTURO}, year={2014}, pages={1–28}}

@article{2013, title={Efficient Estimation of Word Representations in Vector Space}, url={http://arxiv.org/abs/1301.3781}, abstractNote={We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.}, author={Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey}, year={2013}, month={Jan}}

@book{2012, title={Artificial Intelligence: A Modern Approach}, ISBN={9780123969590}, DOI={10.1016/B978-0-12-396959-0.00001-X}, author={Norvig, Peter}, year={2012}}

@article{1–22008, title={Opinion Mining and Sentiment Analysis}, volume={2}, ISSN={1554-0669}, url={http://www.nowpublishers.com/article/Details/INR-011}, DOI={10.1561/1500000011}, number={1–2}, journal={Foundations and Trends® in Information Retrieval}, publisher={Now Publishers Inc.}, author={Pang, Bo and Lee, Lillian}, year={2008}, pages={1–135}}

@article{2002a, title={Thumbs up? Sentiment Classification using Machine Learning Techniques}, url={http://www.aclweb.org/anthology/W02-1011}, abstractNote={We consider the problem of classifying doc-uments not by topic, but by overall senti-ment, e.g., determining whether a review is positive or negative. Using movie re-views as data, we find that standard ma-chine learning techniques definitively out-perform human-produced baselines. How-ever, the three machine learning methods we employed (Naive Bayes, maximum en-tropy classification, and support vector ma-chines) do not perform as well on sentiment classification as on traditional topic-based categorization. We conclude by examining factors that make the sentiment classifica-tion problem more challenging.}, author={Pang, Bo and Lee, Lillian and Vaithyanathan, Shivakumar}, year={2002}}

@article{2018, title={Some features speak loud, but together they all speak louder: A study on the correlation between classification error and feature usage in decision-tree classification ensembles}, volume={67}, ISSN={0952-1976}, url={http://www.sciencedirect.com/science/article/pii/S0952197617302488}, DOI={10.1016/J.ENGAPPAI.2017.10.007}, journal={Engineering Applications of Artificial Intelligence}, author={Ramirez-marquez, Jose}, year={2018}, pages={270–282}}

@article{42013, title={Whose and what chatter matters? the effect of tweets on movie sales}, volume={55}, ISBN={0167-9236}, ISSN={01679236}, url={http://dx.doi.org/10.1016/j.dss.2012.12.022}, DOI={10.1016/j.dss.2012.12.022}, abstractNote={Social broadcasting networks such as Twitter in the U.S. and “Weibo” in China are transforming the way online word of mouth (WOM) is disseminated and consumed in the digital age. In the present study, we investigated whether and how Twitter WOM affects movie sales by estimating a dynamic panel data model using publicly available data and well-known machine learning algorithms. We found that chatter on Twitter does matter; however, the magnitude and direction of the effect depend on whom the WOM is from and what the WOM is about. Incorporating the number of followers the author of each WOM message had into our study, we found that the effect of WOM from users followed by more Twitter users is significantly larger than those followed by less Twitter users. In support of some recent findings about the importance of WOM valence on product sales, we also found that positive Twitter WOM is associated with higher movie sales, whereas negative WOM is associated with lower movie sales. Interestingly, we found that the strongest effect on movie sales comes from those tweets in which the authors expressed their intention to watch a certain movie. We attribute this finding to the dual effects of such intention tweets on movie sales: the direct effect through the WOM author’s own purchase behavior, and the indirect effect through either the awareness effect or the persuasive effect of the WOM on its recipients. Our findings provide new perspectives to understand the effect of WOM on product sales and have important managerial implications. For example, our study reveals the potential values of monitoring people’s intentions and sentiments on Twitter and identifying influential users for companies wishing to harness the power of social broadcasting networks. © 2012 Elsevier B.V.}, number={4}, journal={Decision Support Systems}, publisher={Elsevier B.V.}, author={Rui, Huaxia and Liu, Yizao and Whinston, Andrew}, year={2013}, pages={863–870}}

@article{2010, title={Predicting elections with Twitter: What 140 characters reveal about political sentiment}, ISBN={0894439310386}, ISSN={00219258}, url={http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1441/1852}, DOI={10.1074/jbc.M501708200}, abstractNote={Twitter is a microblogging website where users read and write millions of short messages on a variety of topics every day. This study uses the context of the German federal election to investigate whether Twitter is used as a forum for political deliberation and whether online messages on Twitter validly mirror offline political sentiment. Using LIWC text analysis software, we conducted a content analysis of over 100,000 messages containing a reference to either a political party or a politician. Our results show that Twitter is indeed used extensively for political deliberation. We find that the mere number of messages mentioning a party reflects the election result. Moreover, joint mentions of two parties are in line with real world political ties and coalitions. An analysis of the tweets’ political sentiment demonstrates close correspondence to the parties’ and politicians’ political positions indicating that the content of Twitter messages plausibly reflects the offline political landscape. We discuss the use of microblogging message content as a valid indicator of political sentiment and derive suggestions for further research.}, journal={Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media}, author={Tumasjan, Andranik and Sprenger, To and Sandner, Pg and Welpe, Im}, year={2010}, pages={178–185}}

@article{2002b, title={Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews}, url={http://www.aclweb.org/anthology/P02-1053.pdf}, abstractNote={This paper presents a simple unsupervised learning algorithm for classifying reviews as recommended (thumbs up) or not rec-ommended (thumbs down). The classifi-cation of a review is predicted by the average semantic orientation of the phrases in the review that contain adjec-tives or adverbs. A phrase has a positive semantic orientation when it has good as-sociations (e.g., “ subtle nuances ”) and a negative semantic orientation when it has bad associations (e.g., “ very cavalier ”). In this paper, the semantic orientation of a phrase is calculated as the mutual infor-mation between the given phrase and the word “ excellent ” minus the mutual information between the given phrase and the word “ poor ” . A review is classified as recommended if the average semantic ori-entation of its phrases is positive. The al-gorithm achieves an average accuracy of 74% when evaluated on 410 reviews from Epinions, sampled from four different domains (reviews of automobiles, banks, movies, and travel destinations). The ac-curacy ranges from 84% for automobile reviews to 66% for movie reviews.}, author={Turney, Peter D}, year={2002}}

@misc{2017b, title={Sentiment analysis of tweets with Python, NLTK, word2vec &amp; scikit-learn - Marcin Zabłocki blog}, url={http://zablo.net/blog/post/twitter-sentiment-analysis-python-scikit-word2vec-nltk-xgboost}, year={2017}}

@misc{2017c, title={Spanish Billion Word Corpus and Embeddings}, url={http://crscardellino.me/SBWCE/}, year={2017}}

</bibliography>

<latex>

\bibliography{bib}

</latex>