Fast Opinion Mining using Information Retrieval Techniques

JOSE ANTONIO ORTIZ BASCUAS[1]

Menéndez Pelayo International University,
La Rioja nº 47 28697 Villanueva de la Cañada - Spain
[1]jaorbas@protonmail.com

Abstract. This paper focuses on the construction of models, through automatic learning, for sentimental analysis, which allow obtaining the polarity of a tweet by taking advantage of the information obtained through an information retrieval process. For this purpose, the features derived from the classification generated by such a system in response to the consultation of the document to be analysed will be used. Through this combination of tools, we will achieve a language-independent sentiment analysis, reaching accuracies comparable to other state-of-the-art approaches but at a much higher speed.

Keywords: sentiment analysis, opinion mining, information retrieval, polarity classification, Twitter, intelligent intuition

(Received September 21st, 2020 / Accepted November 17th, 2020)

1 Introduction

Opinion mining (OM), also often referred to as sentiment analysis, is dedicated to the computational study of opinions expressed in text form, i.e., identifying a person's attitude towards a topic or the general contextual polarity (positive or negative) of a document [18].

Social networks such as Twitter are an almost inexhaustible source of opinions on all kinds of subjects and, as such, the object of study of multiple disciplines. In any case, the subjectivity contained in the tweets is a valuable source of information provided by its users.

Why tweets? A tweet is considered a unit of opinion, of which there is a large searchable database. The approach analyzed in this work would not work properly if opinions or arguments were mixed in the same text in several ways. However, in the case of tweets, given their size, the use made of them in this context is to express an opinion as clearly and as narrowly as possible. To a certain extent, a tweet could be considered as a unit of opinion. This, together with their high availability, makes them ideal for this project.

The hypothesis explored in this paper is basically that similar Twitter posts tend to belong to the same class. Therefore, information about the class of their *n* most similar posts can help classify the polarity of a new unlabeled tweet. To implement this solution, the approach analysed in this paper is based on the degree of similarity between publications obtained through an *Information Retrieval System* (IRS).

The objective of this work is to review the design that the system should have, which allows it to exploit this IRS resource to obtain certain characteristics, and its use in the OM on tweets.

Before conducting a review of the OM situation in the context described above (section 3), consideration has been given to including what an IRS is and how it works as an introduction (section 2). In the following sections the hypothesis is presented and describes methods of extracting features compatible with the problems to be solved (section 4), which will be measured and compared with other methods (sections 5 and 6). Finally, the last section (section 7) includes a series of conclusions and other issues that will contextualize this work in the state of the art.

2 Information Retrieval System

Information Retrieval System (IRS) is the science of searching for information in any type of digital documentary collection, and as an objective, it performs the retrieval in text, images, sound or other data features, in a relevant and pertinent way.

To achieve its objective, it relies on information systems techniques are used to automatically determine: the search criteria, the relevance and pertinence of the terms. In practice, the organization of information is done in a data structure called index, from which, once built, queries can be made.

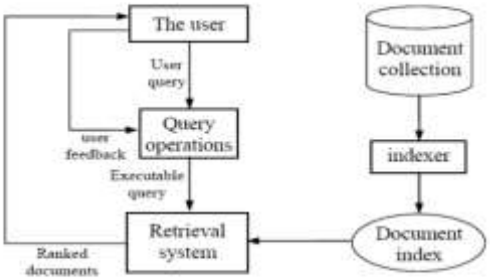

Figure 1. Information Retrieval Architecture.

When a user makes a query to the index, several objects can belong to the answer, although with different degrees of relevance. Most IRS compute a *ranking* to know how well each object responds to the query, ordering the objects according to their ranking value (or similarity). In practice there may be different nuances of relevance, but they always try to return the information (not just data) that is inferred from the user's search terms.

3 Related Work

In recent years, interest has grown in the automatic processing and analysis of opinions expressed by users on social networks, especially tweets, and various approaches have been proposed.

Mostly a *bag-of-words* (BoW) approach has been used in which words in documents are used as features, usually unigrams and bigrams [4, 6, 7, 8], combined with dictionaries of feeling terms [4]. Morphosyntactic (*part-of-speech)* tags have also been widely used as features [2, 4, 6, 7, 10]. In the case of [5], traditional feature selection strategies were examined and a semi-automatic method was proposed to identify those of greatest interest. In this way, they generated a lexicon with 187 features (i.e., terms) derived from a dataset formed by tweets about the singer Justin Bieber.

More recently in [5], following the work of [13], he discusses the use of feature *hashing* to solve the problem of dispersion of vectors created from tweets when words are used as features, i.e. in the case of BoW approaches. In this approach, features are hash integers rather than strings. The authors showed that the feature hashing approach outperformed BoW in the experiments conducted.

In [21] its authors enhanced a standard bigrame model with features that represent contextual information about the tweet: its geographical origin, the time of day, the day of the week, the month and the tweet's author. Their experiments identified a 10% gain in accuracy with the additional features. The limitation of this approach is that such contextual information is not available in the standard data sets used in the literature.

Likewise, in [9] a feature expansion was performed by adding, to a BoW model, features that explore the presence of adjectives, emoticons, emphatic and onomatopoeic expressions or also expressive word lengthening. The authors found that adjectives were the most discriminating features, with gains ranging from 0.49% to 4% accuracy depending on the data set. To obtain these additional features, a POS tagger and a specialized lexicon of feelings are needed.

On the other hand, in the reference to the most popular automatic learning algorithms for sentiment analysis, Multinomial Naïve Bayes, Vector Support Machines and Maximum Entropy stand out [10]. Likewise, recent studies have explored the combination of classifiers. Sets of classifiers and clusters improved the quality of classification, but brought additional computational costs [5, 9].

In the same line of our work, we find the approach proposed by [1], where a automatic fix selection of features for each class was made. Each characteristic is a function (or a simple statistic) that is applied to the ordered list of results obtained by consulting each tweet in an IRS that contains all the tagged tweets under study.

But if we consider the proposals presented in different competitions, for example in TASS 2019. The most common strategies (RETUYT-InCo[1] y GTH-UPM[2]) are

[1] http://ceur-ws.org/Vol-2421/TASS_paper_6.pdf

[2] http://ceur-ws.org/Vol-2421/TASS_paper_3.pdf

based on the use of Deep Learning, in combination (through weighting) with these classic classification methods, from n-grams and other characteristics extracted from the tweet to be classified. For example: number of words in the tweet, number of words with capital letters, up togs and number of positive/negative words (BoW), and others.

The automatic processing of the Semantic Web will be promising, but today it does not seem possible for users to upload semantic metadata with their posts in the social networks (in a language-independent context), beyond the emoticons. As with Deep Learning and word embedding there is a lot of literature and related methods to highlight. They are interesting and essential to mention in the state-of-the-art to have an overview, but the technical differences, performance and context of use do not make them comparable with the approach of this work and could introduce noise in the conclusions. We prefer to limit the scope of our work to the classical classification family methods, still essential.

So, the scope of our work is to leverage and take into account that IRS includes *Semantic Vector Space* funcionalities to improve search engine competence, and use it for feature extraction. This approach is comparable to classic n-gram based methods, still effective in the actual context. But also poorly explored at the moment.

4 Proposed solution

Our goal is to obtain an implementation that allows us to classify the polarity of opinions in a collection of documents. In our case, the features to be exploited will not be the n-grams contained in these documents, as was the case in other approaches already mentioned. On the contrary, what we will exploit will be a sufficient and representative number of features obtained from consulting each tweet in an IRS where the training documents of each *dataset* will be stored and tagged, in order to train in a supervised way a model that allows us to make the most reliable predictions possible. To measure this reliability, we will compare our results with those of other n-gram based methods.

The process of obtaining this model consists of several steps that we will discover throughout this section:

1. Construction of the index.
2. Selection of features.
3. Dimensionality revision.
4. Training.

To do this we must have certain resources: a labelled dataset, an IRS and software that allows us to calculate these models.

From the query of a tweet in the IRS index, we will obtain as output a list of relevant tweets, with the information shown in Table 1: ranking number, meta-information contained in the tweet in question (including the class +/- which tells us if the tweet expresses a positive or negative opinion) and the value of similarity with the query of the input tweet and the index identifier.

Table 1: Result of an IRS query

```
1. 11/+ (score 2.54, docid 1)
2. 34/+ (score 2.14, docid 4)
3. 57/+ (score 1.71, docid 37)
4. 21/- (score 1.62, docid 21)
5. 45/+ (score 1.60, docid 435)
```

The above is a generalization of the hypothesis of [1], whose method they called SABIR, using 12 fixed statisticians for each class. But previous studies on the patterns in the graphic representations of the texts to be classified have allowed us to offer in this work a different starting point.

4.1 Graphic representations of a tweet

Before we continue, we should take a moment to see what information we are going to get, and how to exploit it. The only reference to emotions that is made in the whole process is labelling.

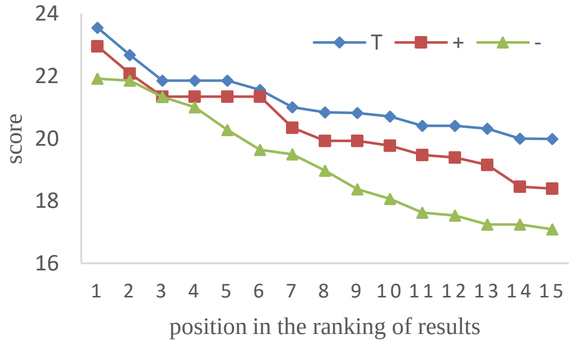

Figure 2. Similarity values obtained for *"had a very productive day today"*.

According to our hypothesis, documents in the same category will have a common pattern that our IRS will be able to identify and from which we will be able to extract the model.

To prove this hypothesis, let's take, for example, the STD data collection (which we will describe in more detail in section 5) and build from it three indexes: one for all those tweets labeled with positive polarity ("+"), one for those labeled with negative polarity ("-") and one index with all the tweets of both classes ("T"). What will happen when we look for our tweets in each of the three indexes? Since for each query we obtain a decreasing series of similarity values with respect to the indexed tweets in the queried index, we can represent them graphically and compare them to obtain some answers. Obviously, for the construction of the indexes, we will only use the training data when we intend to build a model.

Based on a positively polarized example tweet, Figure 2 shows the similarity values corresponding to the first 15 tweets returned by the IRS for each of our three indices. The same happens in Figure 3 for the negative polarization tweet and in Figure 4 for a neutral tweet.

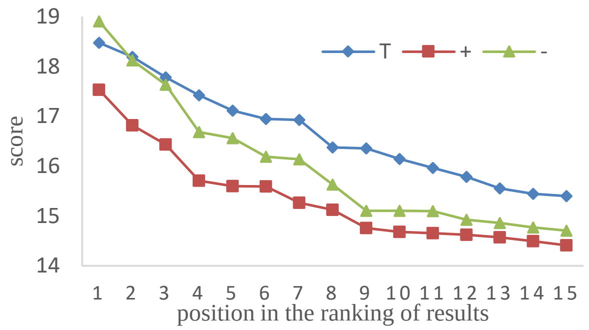

Figure 3. Similarity values obtained for *"Felling fat after a day off with food & drink"*

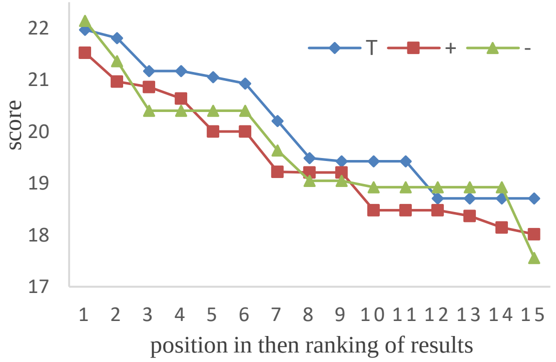

Figure 4. Similarity values obtained for *"just landed at San Francisco"*.

A pattern compatible with the hypothesis can be seen in Figures 2, 3 and 4. The lines corresponding to the similarity values obtained for each index appear clearly "stratified" when the tweet has some degree of polarity, especially if we compare it with the case of the tweet labelled "neutral". If the tweet is positive, the similarity values in the index query "+" will obtain higher values than those obtained for the index query "-", and vice versa. Sometimes "confusion" can be observed represented as the crosses between lines of the tweet graph that prevent a conclusive interpretation, and that appear when the polarity of the tweet decreases. Simply calculating the area under both lines to derive a model is somewhat naive, so we will need to extract more features to include.

Thus, by consulting the "T" index, we will obtain the scores and labels of the documents that most resemble the tweet we want to label. And presumably, if our hypothesis is correct, we will obtain information (from the data retrieved from the IRS) for the classification that will help us decide on the prediction of one class or another for the tweet in question. For example, if all the most similar recovered tweets are positive and have a high similarity value, can the tweet under study be negative? In addition, we can obtain a measure of belonging to each polarity or class if we consult the same tweet in the indexes built for each class, which if posed alone do not contribute anything until we compare their values.

The question is, therefore, to find feature extraction functions that are sensitive to the order and value of the returned tweets after consulting the "T" index, and to define which features we can extract from a tweet query in the "+"/"-" indexes. That is, from how many different points of view can we measure the belonging of each tagged tweet to each class? All the indices involved have different and independent information about the polarity of a tweet since they contain different indexing terms.

The pattern in Figure 4 is met for 20-30% of the elements included in the previous study. It varies greatly depending on the quality of the classification and the theme of the data set.

4.2 Aggregation features

Once our working hypothesis has been set out, let us now analyse several aspects to be taken into account when building our model. The first of these is the aggregation features.

An *aggregation function* is a function that takes a list entry and results in a value. In this way, we can generate a function like:

$$\operatorname{avg}(x_1, .., x_n) = \frac{n_+ - n_-}{n} \tag{1}$$

where *n is* the number of responses after the IRS query, *n+* is the number they have positive polarity, *n- the* tweets with negative polarity. It could be considered a kind of average, if we assign a value of +1 or -1 depending on the polarity of the tweet.

Another feature, with a lot of information for our classifier, will be the one that measures the similarity value according to the relative position of the ranking it occupies according to its class:

$$\text{rank}(x_1, .., x_n) = \sum_{k=0}^{n} \frac{rank_{rel}}{rank_{abs}} \cdot score(x_k) \cdot polarity(x_k) \tag{2}$$

where the *polarity* function expresses the sign of the labeled polarity of each IRS response. $rank_{rel}$ refers to the position of the item within the list of its class, ignoring items from other classes. And $rank_{abs}$ is the position that the item occupies as it has been returned from the IRS. Thus, it is worth +1 if the polarity is "+" and has a value of -1 if it is "-". By including the value for both classes, we will facilitate the work of the training; whatever method is chosen.

Table 2: IRSACAgr features obtained from Table 1

avg	rank	$Area_+$	$Area_-$	class
0,6	7,63	10,05	6,44	+

To complete this selection of features obtained only from consultations to the "T" index, we can include the area of the curve resulting from the graphic representation of each tweet after consulting it in each index, as is the case of figures 2, 3 and 4. These areas are no more than the sum of the similarity values after consulting the "+" (Area+) and "-" (Area-) indexes.

Thus, reflected in Table 2, if we assume that the tweet consulted is labeled with '+', we would obtain for those features the values shown in Table 1, proceeding in a similar way with each tweet labeled in our index.

This would generate a list of values that could already be included in an automatic learning process. The features have been chosen so that they do not match with any of those already included by [1], and thus build a model based on new features. We will call this method IRSACAgr, but the user can generate his own with more specific functions for the problem he wants to solve.

4.3 Ranking features

Unlike the aggregation features, in the *ranking features* will select a number *n* of response items, and for each of them we will implement a function with its similarity value and metadata as input parameters.

For example, in the case of the previous section, we will use as a characteristic the same value (absolute) of similarity, but with negative sign for the polarity tweets "-" and positive sign for the polarity tweets "+". We will take a fixed number *n* of answers. If the number of results were higher, all those with a ranking higher than *n would be* ignored*;* and if it were lower, the missing values would be filled in with 0. This time, for the line in the example, and n=5, we would get the values shown in Table 3:

Table 3: RSACRank features obtained from Table 1 for n=5

Index	C1	C2	C3	C4	C5	class
T	2.54	2.14	1.71	-1.62	1.60	+
+	3.20	2.52	1.57	1.55	1.21	+
-	-2.68	-1.55	-1.55	-0.43	-0.23	+

As it was done for the aggregation features, we can also take advantage of the information for the classification provided by the query, from the same tweet, to the "+" and "-" indexes, thus obtaining *n*3* features which we will include for the same instance for the training.

We will call this method IRSACRank.

4.4 Ranking features

However, the guidelines given for classifying an opinion as positive ("+") or negative ("-") would be incomplete if they did not give the possibility of extending our proposal to a multi-class solution.

Many times the labels of the tweets indicate degrees of polarity; or they appear classified as neutral polarity ("NEU"), if the opinions expressed point with equal intensity to both directions of polarity; or even as null polarity ("NULL"), if the tweet does not express any opinion in any sense (although we will consider them as *outliers* and they will not be included in any index).

Moving our hypothesis to this new context, in the case of tweets with neutral polarity, they should be included in both "+" and "-" indexes, for training, since they have some of both signs (much or little alike). On the contrary, they should not be included in the "T" index, since from this index features are extracted that help to discriminate one of the signs of polarity. If it were included, its influence would modify the similarity values and features, when it should have no influence at all. Intuitively it can be seen that, if both feelings are countered, in IRSACAgr the values of Area+ and Area- will be similar, while those of rank and avg will be low. Something similar will happen for IRSACRank. These are therefore clues that the chosen machine learning method will have to identify in order to make a prediction.

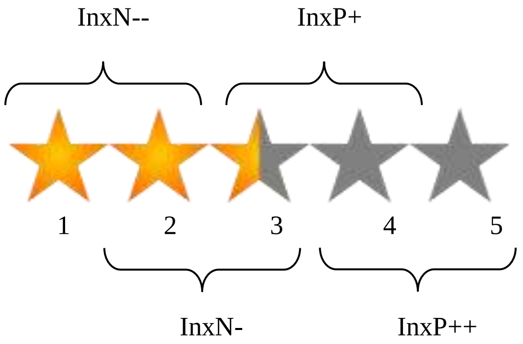

Figure 5. Indexing of labels associated with a 5-star opinion

To illustrate this, let's take the collection of opinions from Amazon users (dataset later referred to as AMZ), who in addition to their opinion, have labeled the product with between 1 and 5 stars, indicating their level of satisfaction. The resulting number of classes is now more varied and also includes a "NEU" class, which we could associate with a neutral polarity, which would correspond to those opinions labeled with 3 stars.

In this case, we will have a "T" index that will contain all the opinions except those labeled with 3 stars (class "NEU"). Also, instead of an index for each class, we will build four overlapping indices so that the "NEU" class opinions are included in two of these indices, as shown in figure 5.

This new class organization differs from those described for defining IRSACAgr, since only two polarities were contemplated. The initial Area+ and Area- aggregation functions are now four (AreaInxN--, AreaInxN-, AreaInxN+, AreaInxN++), one for each class grouping we have considered. The rank aggregation function also changes since now polarity will not be valid -1 or +1 (valid only for two polarity values), but it will be necessary to include intermediate values according to their polarity intensity and their sign. Thus: polarity("1 star") = -1, polarity("2 stars") = -0.5, polarity("3 stars") = 0, etc.

And something similar occurs with IRSACRank, where n features will be extracted from the now 5 indices, therefore each instance will have *n*5* features. And from the features extracted from "T", now it will not be enough to change its sign according to its polarity, but the *polarity* function, previously described, will be reused to obtain the weight of the characteristic (see Preprocessing in Section V) according to its labeling.

4.5 Dimensionality review

In the case of the aggregation features we can include all the functions that we think may be relevant. In this case, reviewing their dimensionality would be part of the exploratory analysis of the data to help understand them. We have applied PCA analysis to choose those features that accumulate the most information.

Ranking features are a list of *scores* of the tweets returned by an IRS and the lower the similarity value the less information it provides. As it is a descending orderly list, it will be enough to choose a maximum value for *n* (number of items returned in the index query) that will give us the information we need for the training. Fortunately, the performance offered by an IRS for the extraction of a reasonable number of features is good, so we can choose a high number and then perform the dimensionality study.

No comprehensive study has been done, but for reference, loading a rate of 1.6 million tweets takes about 8.7 seconds on the computer used for all of our: an Intel® Core™ x64 i5-7300U CPU @ 2.60GHz 2,70 GHz RAM 8.00 GB and, despite the volume, query times are almost negligible. This is not the case, when using them for model calculation, as processing time increases exponentially with *n* as will be seen later.

5 Evaluation

We then proceed to evaluate our proposal. First, we will describe our experimental environment.

Dataset. It is made up of six collections of real tweets and comments, previously used by the research community in tasks of this type (you can see quantitative details of the datasets in figure 4):

- **TASS2014.2c (TA^2) y TASS2014.3c (TA^3) Task 1: Sentiment Analysis at global level**. This is a collection of user reviews of various products and services, such as those posted on TripAdvisor[3] to comment on a restaurant or hotel, for example. The labelling has four classes (P, N, NEU, NONE). Those labeled "P" and "N" have been indexed to build TASS2014.2c and "NEU" has been included to build TASS2014.3c.

- **Stanford-Twitter sentiment corpus[4] (STD)**. The original dataset is made up of 1.6 million automatically tagged tweets, we will only use 30,000 (due to hardware limitations). Its tagging is noisy, and basically consists of assigning a polarity to the tweet based on the feelings associated with the emoticon present in the tweet.

- **Amazon product data (2018):[5] Music Instruments** (**AMZ**). This data set contains positive and negative feedback (5 tags according to the star rating) for thousands of Amazon software products. A selection has been made of those items included in the dataset that do not exceed 1024 bytes in length due to IRS limitations.

- **Twitter US Airlines Sentiment[6] (AIR).** Presented on the Kaggle platform and relating to traveler issues with U.S. airlines. The Twitter data was obtained from February 2015 and the scorers were asked to first rank the positive, negative and neutral tweets.

- **Real or Not? NLP with Disaster tweets[7] (DTE)**. This peculiar Kaggle challenge aims to identify which tweets are about real disasters and which are not. It includes a collection of 10,000 hand-tagged tweets with two classes ("Yes" if it's a real disaster and "No" otherwise). This dataset has been included to check whether our hypothesis can be extended to other types of tagging, and to answer in this case the question; do tweets that deal with a real disaster (or not) tend to belong to the same class?

Pre-processed. Only the double quotes have been replaced by their escaped value because they have a special meaning for the IRS search engine used. Line breaks have also been eliminated.

Classification tools and methods. The classifiers implemented and integrated in the Weka[8] framework (Hall et al., 2009) in its version 3.8 have been used, and for each classification method (or variable selection) they have been used with their default parameters.

Sections have been extracted from the original files (15% as they were considered to be large enough) to build test files. Using the remaining 85% for training.

In addition, all tests have been performed at a similar level of system load so that the data obtained are comparable.

Table 4: Details of the composition of the datasets

Set	#items	classes (K)	#terms	#indx
AIR	14.631	9/3/2.3	256k	14k
AMZ	11.611	1.4/0.6/1.4/3/5	1.647k	32k
STD	30.000	15/0/15	393k	37k
TA^2	5.059	2.2/2.9	99k	17k
TA^3	5.728	2.2/0.7/2.9	99k	17k
DTE	7.608	4.3/3.2	51k	10k

Evaluation metrics. The standard evaluation metrics used in classification systems have been used: *accuracy (*Acc), *precision (*Pr), *recall (*Rec), *F-Measure* (F-M), and AUC. Since these are very small, paired samples and it cannot be proven that they have a normal distribution, the Wilcoxon test is used to see if the difference in the results obtained with respect to the baseline is significant, compared with IRSACAgr and IRSACAgr.

IRS system. Zettair [9](GPL license) has been chosen because it is a simple IRS, with great performance, command line operability and flexibility in its configuration. The Okapi BM25[10] similarity metric (k1 = 1.2, b = 0.75) has also been used. No further adjustments of the IRS configuration have been taken into

[3] https://www.tripadvisor.es/

[4] http://help.sentiment140.com

[5] https://www.kaggle.com/eswarchandt/amazon-music-reviews?select=Musical_instruments_reviews.csv

[6] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[7] https://www.kaggle.com/c/nlp-getting-started/data

[8] https://sourceforge.net/projects/weka/

[9] http://www.seg.rmit.edu.au/zettair/

[10] https://nlp.stanford.edu/IR-book/html/htmledition/okapi-bm25-a-non-binary-model-1.html

consideration to optimize the results in *score* or response time.

Baselines. SVM has been chosen for automatic learning based on the features obtained from the IRS: SABIR, IRSACAgr and IRSACAgr. For the "classic" methods (SVM and Naïve Bayes), Weka has been used, and n-grams of size 1 to 2 have been extracted by means of StringToWordVector with an IteratedLovinsStemmer for the normalisation of the terms and a selection of attributes based on *InfoGainAttributeEval*. Only these methods have been included because they are well known and still widely used. These methods have been included because they are well known and still widely used. We consider that, other methods (like Deep Learning family methods) could introduce noise in the conclusions because they may belong to very different contexts of use.

5.1 Dimensionality for IRSACRank

Before executing the experiments on the different datasets, an important detail remains to be defined, the dimensionality for IRSACRank. The value of *n* for IRSACRank is still unknown, and it is not yet known whether the performance of the chosen features, on the results obtained, depends on the dataset.

If we represent the value of the AUC metric as a measure of the model's bonanza and as a function of the number of features (which in IRSACRank is equivalent to the number of items recovered from IRS in each query) we obtain a performance as shown in Figure 6. That is, from which response item, the information that provides the performance stops improving (other classifiers and metrics have been tried, with a similar results).

Therefore, this value *n* needs to be calculated to provide sufficient information to calculate a reasonable model (figure 6) but without triggering the time taken to calculate the model (figure 7).

6 Results

In Table 5 we compare the results obtained from each dataset with the classification methods based on n-grams and those based on IRS features.

It has been chosen not to indicate the precision in those cases where there are classes for which the model has not made any predictions. This only occurs in those datasets with several polarity classes, or that includes a neutral class.

There are several models that could be discarded because by, offering an AUC of almost 0.5, they would be no better than a ZeroR classifier. In particular, this is the case with the AMZ dataset for all methods except Naïve Bayes although it doesn't offer better accuracy either. The reason is that training datasets with unbalanced classes can generate unstable models. The dataset offers very long comments, which allows for the inclusion of opinions with different polarities, and as a consequence indexing terms in the contaminated class indices, which does not allow for proper class discrimination. For the rest of cases, the models behave reasonably well.

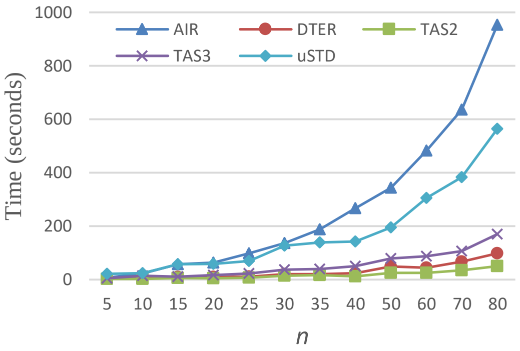

Figure 7. Time needed to calculate the models represented in Figure 6 according to the value of *n*.

Table 5: Classification method performance (highlighted results with better dataset accuracy). Wilcoxon hypothesis contrast on the accuracy between each method with (a) IRSACAgr and (b) IRSACAgr for a significance level α=0.05 and H0: μ=0.

		TA^2	TA^3	AIR	AMZ	STD	DTE
SVM	***Acc.***	**,732**	**,661**	**,767**	**,697**	**,755**	**,792**
(a) H_a: 0 < α (b) H_a: 0 < α	*Pr.*	,733	,613	,756	-	,758	,794
	Rec.	,732	,661	,767	,697	,755	,792
	F-M	,727	,631	,755	-	,754	,788
	AUC	,716	,702	,772	,543	,755	,773
N. Bayes	***Acc.***	**,687**	**,583**	**,712**	**,621**	**,682**	**,703**
(a) H_a: 0 < α (b) H_a: 0 < α	*Pr.*	,688	,579	,710	,651	,682	,706
	Rec.	,687	,583	,712	,621	,682	,703
	F-M	,688	,581	,710	,634	,682	,704
	AUC	,754	,719	,835	,719	,741	,767
SABIR	***Acc.***	**,766**	**,683**	**,758**	**,704**	**,743**	**,813**
(a) H_0: .49 ≥ α (b) H_0: .49 ≥ α	*Pr.*	,766	-	,747	-	,743	,817
	Rec.	,766	,683	,758	,704	,743	,813
	F-M	,766	-	,741	-	,743	,809
	AUC	,761	,709	,753	,505	,743	,794
IRSACAgr	***Acc.***	**,755**	**,685**	**,710**	**,704**	**,745**	**,813**
	Pr.	,754	,665	,700	-	,745	,814
	Rec.	,755	,685	,710	,704	,745	,813
	F-M	,753	,657	,640	-	,745	,810
	AUC	,746	,713	,668	,503	,745	,797
IRSACRank	***Acc.***	**,748**	**,686**	**,742**	**,703**	**,745**	**,811**
	Pr.	,747	,668	,725	-	,745	,812
	Rec.	,748	,686	,742	,703	,745	,811
	F-M	,746	,667	,724	-	,745	,808
	AUC	,752	,708	,753	,503	,745	,794

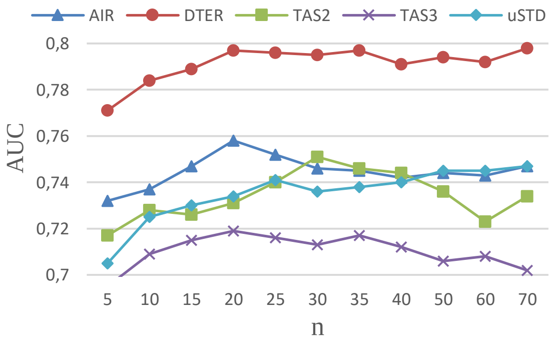

Figure 6. AUC value for the SMO classifier, by the extraction of *n* features described for IRSACRank

An important difference from IRS-based methods is the time of feature extraction (see Table 6, where is included the time spent building indexes for IRS-based methods and Weka's toWordVector + AttributeSelection for n-gram-based methods.) with which the model is subsequently trained. This difference is statistically demonstrated with a p-value of approximately 1. Likewise, for the time spent in the calculation of the model, only Naïve Bayes surpasses IRSACRank with a similar p-value. However, IRSACAgr outperforms all the others, needing less than a second for all the tests of interest, offering without impairment, an *accuracy* similar to the rest.

Table 6: Costs in time (seconds) of feature extraction (F) and model calculation (M).

		TA^2	TA^3	AIR	AMZ	STD	DTE
SVM	*F*	517	664	1.721	>5k	4k	902
	M	13,4	25,5	158,2	139	657	34,2
N. Bayes	*F*	517	664	1.721	>5k	4k	902
	M	3,8	3,4	10,7	3.57	29,2	8,3
SABIR	*F*	29	37	76	167	201	45
	M	0,7	12.0	12.6	248	23.5	2.2
IRSACAgr	*F*	23	29	81	100	164	44
	M	0.1	0.2	0.5	2.1	0.6	0.1
IRSACRank	*F*	36	45	77	146	230	48
	M	7.5	21.1	65.2	541	200	14.7

7 Conclusions

We have assumed that documents in the same category will have a common pattern, and the features needed to build the model can be extracted with the help of an IRS. To this end, we have constructed an index for each class, to measure the belonging of an opinion to each one of them, and an index that includes all the opinions to contextualize an opinion that helps us to discriminate some classes against others. We have built two methods for extracting features (IRSACRank and IRSACAgr) for application in Opinion Mining and Sentiment Analysis tasks. What characterizes these methods is that the features they use are extracted from the described indexes through an Information Retrieval process. Both generate models of similar performance, although the first of them, IRSACRank, uses a collection of "raw" data that makes many variables to be processed in the calculation of the model, and is more demanding in terms of hardware resources. However, it has a very interesting advantage. IRSACrank includes all the information needed by IRSACAgr, to calculate its aggregation values functions and generate the few variables it needs (either those already defined in this work or other new ones that you want to incorporate), which makes it much lighter, ideal for high performance tasks or online. Both models are complementary. In this sense IRSACRank will define the extraction of all data that IRSACAgr can convert into a summary through the aggregation functions that it implements.

The methods described are compatible with other improvements, such as extending the metadata part included in the index, to give more information from de dataset. Each problem to be solved may require an adaptation to improve its modeling. Furthermore, as IRC is capable of indexing unstructured content, with the techniques explained, it would be possible to implement, for example, fast and simple voice recognition systems.

According to our results, no one method outperforms the others for all the measurements and datasets used. Therefore, it is necessary to understand the problem and the distribution of the data (and opinions) in order to choose the method that works best, and in particular, the number of features extracted from the Information Retrieval process to be used. As our experience shows, using more features does not guarantee better results.

We take for granted the fact that similar opinions tend to belong to the same class when we index them in an IRS. Although we have also seen, by including the DTE dataset, that these do not necessarily have to be opinions, but that the same subjectivity with which the labeling was done, if the same criteria were maintained for the whole dataset, will produce similar results. It should also be noted that models obtained from one data set do not accurately predict the response of data from another data set with different themes.

Relevant aspects remain to be debugged, such as the instability measured in the classification methods when the neutral class was introduced or a polarity gradation, although we leave it as future work since it is outside the scope of this project.

References

[1] A.U. Kauer & V.P. Moreira (2016) "*Using information retrieval for sentiment polarity prediction*", Ed. Elsevier.

[2] Bakliwal, A., Arora, P., Madhappan, S., Kapre, N., Singh, M., & Varma, V. (2012). *Mining sentiments from tuits. In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis.* WASSA '12 (pp. 11–18).

[3] Barbosa, L., & Feng, J. (2010). *Robust sentiment detection on twitter from biased and noisy data. In Proceedings of the 23rd international conference on computational linguistics: Posters. COLING '10* (pp. 36–44).

[4] Carvalho, J., Prado, A., & Plastino, A. (2014). *A statistical and evolutionary approach to sentiment analysis. In International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - volume 02*. WI-IAT '14 (pp. 110–117).

[5] Coletta, L., da Silva, N., Hruschka, E., & Hruschka, E. (2014). *Combining classification and clustering for tuit sentiment analysis. In 2014 Brazilian Conference on Intelligent Systems* (BRACIS), (pp. 210–215).

[6] Davidov, D., Tsur, O., & Rappoport, A. (2010). *Enhanced sentiment learning using Twitter hashtags and smileys. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. COLING '10 (pp. 241–249).

[7] Deng, Z.-H., Luo, K.-H., & Yu, H.-L. (2014*). A study of supervised term weighting scheme for sentiment analysis*. Expert Systems with Applications, 41 (7), 3506–3513.

[8] Devlin, J., Chang, M-W., Lee, K., Toutanova, K. (2018). "*BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*". arXiv:1810.04805v2 [cs.CL].

[9] Fersini, E., Messina, E., & Pozzi, F. (2016). *Expressive signals in social media languages to improve polarity detection. Information Processing & Management*, 52 (1), 20–35.

[10] Das, Manoj & Padhy, Binayak & Mishra, Brojo. (2017). *Opinion mining and sentiment classification: A review.* 1-3. 10.1109/ICISC.2017.8068637.

[11] Go, A., Bhayani, R., & Huang, L. (2009). *Twitter sentiment classification using distant supervision*. *CS224n project report*. Stanford 1, 12.

[12] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). *The Weka data mining software: An update. ACM SIGKDD Explorations Newsletter*, 11 (1), 10–18.

[13] Lin, J., & Kolcz, A. (2012). *Large-scale machine learning at Twitter. In Proceedings of the 2012 ACM*

SIGMOD International Conference on Management of Data. SIGMOD '12 (pp. 793–804).

[14] Martínez-Cámara, E., Martín-Valdivia, M. T., Urena-López, L. A., & Monte- jo-Ráez, A. R. (2014). *Sentiment analysis in Twitter. Natural Language Engineering*, 20, 1–28.

[15] Mostafa, M. M. (2013). *More than words: Social networks' text mining for consumer brand sentiments. Expert Systems with Applications*, 40 (10), 4241–4251.

[16] Nicholls, C., & Song, F. (2010). *Advances in artificial intelligence: 23rd Canadian Conference on Artificial Intelligence, Canadian AI 2010. In Comparison of feature selection methods for sentiment analysis* (pp. 286–289) .

[17] O'Keefe, T., & Koprinska, I. (2009). *Feature selection and weighting methods in sentiment analysis. In Proceedings of 14th Australasian Document Computing Symposium*.

[18] Pang, B. & Lee, L. (2008). *Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval*, 2 (1-2, Jan.), 1–135.

[19] Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). *Contextual semantics for sentiment analysis of twitter. Information Processing & Management, 52 (1), 5–19. Emotion and Sentiment in Social and Expressive Media.*

[20] Speriosu, M., Sudan, N., Upadhyay, S., & Baldridge, J. (2011). *Twitter polarity classification with label propagation over lexical links and the follower graph. In Proceedings of the First Workshop on Unsupervised Learning in NLP. EMNLP* '11 (pp. 53–63).

[21] Vosoughi, S., Zhou, H., & Roy, d. (September 2015). *Enhanced Twitter sentiment classification using contextual information. In Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 16–24).