Sentiment Analysis

Pedro Leal Varela
Instituto Superior Técnico
pedrolealvarela@me.com

Abstract—Text Categorization is a time-intensive activity that usually requires knowledgeable human experts. A particular area of Text Categorization aims at classifying a document as either conveying a positive or negative feeling - Sentiment Analysis.

In the last decade much work has been done in Sentiment Analysis, especially in the English language due to the large availability of resources (corpora and lexical tools). Currently open problems include the identification of the classification and feature selection method achieving the best performance and whether it is possible to adapt a classification system that is language-independent. This thesis focuses on the latter problem.

To that end, a labeled corpus in Spanish and an unlabeled corpus in Portuguese of movie reviews were built.

Several feature selection methods were tested, to assess their impact on the performance of the classifiers. Supervised and semi-supervised methods were proposed to perform cross-lingual adaptation, in order to make use of English corpora and thus improve classification in other languages.

Classification was made at several levels of granularity, and the results showed that classification of short documents perform better with the Multinomial Naive Bayes classifier and bigrams as characteristics, and longer reviews with Support Vector Machines (SVM) (especially the proposed Weighted SVM) and unigrams. Classification at sentence level showed that sentences are informative about the polarity of the document. Semi-supervised methods were not as effective as the supervised Cross-Lingual method using English data to classify in other languages.

Index Terms—Sentiment Analysis, Feature Selection, Classification, Cross-Lingual Adaptation

I. INTRODUCTION

TEXT Categorization is the task of assigning a Boolean value to each pair \( (d_j, c_k) \in D \times C \), where \( D \) is a set of documents and \( C = \{c_1, c_2, \ldots, c_{|C|}\} \) is a set of predefined categories. A value of \( \top \) assigned to \( (d_j, c_k) \) indicates a decision to file \( d_j \) under \( c_k \), while a value of \( \bot \) indicates a decision not to file \( d_j \) under \( c_k \). More formally, the task is to approximate the unknown target function \( \Phi : D \times C \rightarrow \{\top, \bot\} \) (that describes how documents ought to be classified) by means of a function \( \hat{\Phi} : D \times C \rightarrow \{\top, \bot\} \) called the classifier (rule, hypothesis, or model), such that \( \Phi \) and \( \hat{\Phi} \) “coincide as much as possible” [1].

In order to train a classifier, one needs first to have a training set, \( T^* \). A \( T^* \) is a set of documents in which the class to which it belongs is known (in its most general case, however it may not be so). After the classifier learns from the documents in \( T^* \), it will be able to categorize future documents, making generalizations about the relationship between document content and categories.

A particular task of Text Categorization is Sentiment Analysis, which classifies objects in respect to the general “feeling”. A common task is to classify user reviews of the most variate products (movies, books, electrical appliances, etc.). This sentiment classification task may be either binary or multi-classed. In this work there is only binary classification, with classes defined as positive and negative.

Prior to classification, the documents have to go through a pre-processing step that transforms the text string representation into a numeric feature vector. In this context a feature is a word, number or symbol retrieved from the document and its value, and it is represented by \( w \). A transformation is the bag-of-words that albeit being oblivious to the order of the words in the document, achieves satisfactory accuracy in most topic categorization applications. There are several methods to select the most relevant features given a particular task or a classifier. This process is called Feature Selection (FS).

Sentiment Analysis presents some difficulties, as [2] pointed out: “the whole is not necessarily the sum of the parts”. This becomes evident if, for instance, a document is excessively sarcastic hence resulting in an overall positive aggregate, while in reality the document classification should actually be negative. This work will analyze movie reviews corpora because they provide good material for analyzing subjectivity and opinions from the authors. Movie reviews have been used before for Sentiment Analysis [2], [3]. These reviews are retrieved from the Internet, and are subject to misspelling, either voluntary, to express a kind of stress and intonation (such as haaauuage), or naturally involuntary. One major drawback in Sentiment Analysis is that there are several English corpora available, however, it is hard to build a corpus in other languages, mainly due to lack of resources. One solution is to take advantage of all the English databases freely available on the Internet, and train/test classifiers to be used in other languages. A language independent method may be reached with some cost, especially if made in a automated way. This work is concerned mainly with non-English language analysis, focusing on Portuguese and Spanish.

II. FEATURE SELECTION

FS is a process that selects a subset of features, where the optimality of the subset is measured by an evaluation criterion. High dimensionality is one side of the problem. When there are many extraneous features, learning generally becomes harder, as the model is likely to find spurious relationships between extraneous features in the \( T^* \) that do not generalize well. A high-dimensional \( T^* \) may also lead to sparse data, which will make it difficult for some classifier to find non-zero points and identify a suitable learning function. FS is also effective preventing overfitting [1]. Most of the FS methods are statistically defined by the relationships between the feature,
$w$, and category, $c_k$, and are represented in Table I. A good FS method can make even the simplest classifier model get a satisfying performance through training [4]. Most FS methods will involve the same basic procedure: assign a weight to each feature, score the features based on their weight and retain only a specified number of features, or feature with value, within a pre-specified range. For FS methods that are category dependent, after the computation of the score for each category, there are two popular methods to determine the final score: average, $FS_{avg}(w)$; and maximum, $FS_{max}(w)$ given, respectively by, $FS_{avg}(w) = \sum_{k=1}^{\vert C \vert} P(c_k)FS(w, c_k)$ and $FS_{max}(w) = \max_{k=1}^{\vert C \vert}FS(w, c_k)$. Here are presented the descriptions of the methods that performed best in this work.

### A. Document Frequency (DF)

DF counts the number of documents in the $T^+$ in which a feature occurs, returning the cumulative value. Features with low or high DF are known as rare (if it only appears once it is an unique feature or a hapax legomona) or common features, respectively. It is defined as [5]

$$DF(w) = A + B$$

### B. Mutual Information (MI)

MI measures the amount of information known for classifying a document given that a particular feature exists in the document. The MI between $w$ and $c_k$ is estimated as [5],

$$MI(w, c_k) = \log \frac{N \times A}{(A + B) \times (A + C)}$$

MI assumes that features with high category ratio are more effective for classification. Rare features will have a higher score than common features, which is a weakness of this method.

### C. Information Gain (IG)

In Text Categorization tasks it is defined as the amount of information about a category that is gained by the presence/absence of a feature. It is estimated as [6]

$$IG(w) = - \sum_{k=1}^{\vert C \vert} \frac{N_k}{N} \log_2 \frac{N_k}{N} + \sum_{k=1}^{\vert C \vert} \frac{A}{A + B} \log_2 \frac{A}{A + B} + \sum_{k=1}^{\vert C \vert} \frac{B}{C + D} \log_2 \frac{B}{C + D} + \sum_{k=1}^{\vert C \vert} \frac{C}{C + D} \log_2 \frac{C}{C + D}$$

Comparing with MI, IG makes use of information about absent features, which MI ignores and, IG normalizes the MI scores. It has a natural value of zero for non-informative features, and values increase for features that correlate strongly with certain categories. It has the limitation of being biased in favor of very frequent features.

### D. Cross Entropy for Text (CET)

Unlike IG, where an average of all possible values for a feature is computed, CET only uses the value denoting that the feature occurred in a document. CET is estimated as [7]

$$CET(w) = DF(w) \sum_{k=1}^{\vert C \vert} A \log_2 \frac{A}{N}$$

### E. CHI

When applied to Text Categorization, CHI statistics is used to measure the degree of independence of a feature, $w$, and a category, $c_k$. It is estimated as [5],

$$CHI(w, c_k) = \frac{N \times (AD - BC)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

A feature that occurs frequently in few categories will have a higher value, denoting that it is highly independent. Unlike MI, the values achieved with CHI are cross comparable for the same category, since there is no discrimination of rarity. However, if any of the counts ($A$, $B$, $C$ or $D$) is much smaller than the others this advantage is lost, and there can no longer be a representation by the CHI-squared distribution. Low-frequency features should be removed in order to safeguard against this unreliability.

### F. Odds Ratio (OR)

OR will only take into account features from documents from a given category, and is estimated as [8],

$$OR(w, c_k) = \log \frac{AD}{CB}$$

Values greater than 1 indicate that a feature belongs to a category, otherwise not. This method favors features that occur in positive examples rather than in negative examples, since it does not consider documents that do not contain $w$. A feature that is rare in positive examples, but never occurs in the negative examples will have a relatively high score, despite not necessarily being a good predictor of the category.

### G. Gini Index (GINI)

A value of 0 denotes the minimum of the function, which means that the information since all the information is all on the same category. If all the samples are equally distributed amongst categories then the Gini value is maximum. The Gini Index Text (GIT), introduced in [9], was created in order to work around the Gini Index limitations. It is defined a

$$Ginitext(w, c_k) = \frac{A^2}{\log_2 DF(w)}$$
H. Weighted Log Likelihood Ratio (WLLR)

WLLR was proposed by [10] as an effective FS metric for Text Categorization, and is estimated as

\[ \text{WLLR} = \frac{A}{N_k} \log \frac{A(N - N_k)}{CN_k} \]

A feature will have a high score with respect to a category if it occurs frequently in that category and infrequently in other categories [11]. As demonstrated in [12] WLLR is biased towards features with both high category ratio and high DF.

I. Weight of Evidence of Text (WEIGHT)

WET is based on the average absolute weight of evidence. It was proposed in [13] and it captures the difference between the probability of \( c_k \) occurring given the \( n_{th} \) feature value and the probability of \( c_k \). It is estimated as

\[ \text{WET}(w) = \sum_{k=1}^{[C]} \text{DF}(w) \frac{N_k}{N} \log \left( \frac{A(N - N_k)}{B N_k} \right) \]

J. GU Metric (GU)

[14] introduced this method that uses the z-score to measure the difference in proportions between documents that contain the feature \( w \) and belong to \( c_k \), and those that contain \( w \) and belong to \( c_k \). The larger the z-score, the greater the difference (in proportion) so the feature is better as a discriminator of the two classes.

\[ \text{GU}(w, c_k) = |z| \frac{A(N - N_k)}{B N_k} \]

where \( z \) is defined as,

\[ z = \frac{A - B}{\sqrt{p(1-p)(\frac{1}{N_k} + \frac{1}{N-N_k})}} \]

\[ p = \frac{A + B}{N} \]

III. CLASSIFICATION

The learning approach used by the classifier depends on the knowledge of the \( T^* \). There are three learning approaches: supervised learning, in which the correct label is known for each input of a given \( T^* \); unsupervised learning that finds groupings of similar objects (based on some similarity measure) and does not require \textit{a priori} labeling of training data; and semi-supervised learning that learns from both labeled and unlabeled data for training, using unlabeled data either to modify or re-prioritize hypotheses obtained from labeled data alone.

A. Naive Bayes (NB)

NB is very popular in Text Categorization, owing to its fast and easy implementation. It is based on Bayesian classification, and requires a labeled \( T^* \). The term Naive refers to the assumption that all features are conditionally independent from each other. A direct outcome is that the position of features within a document has no effect in general probability. NB makes a probabilistic model of data within each class, using the \( T^* \) to estimate parameters. These estimates will then be used to classify the test set, \( T^* \), using Bayes rule to compute the posterior probability that the test document was generated by the chosen class and assign it to the class with the highest probability. Even though it is widely used due to its simplicity, it has been targeted with criticism, due to its assumption not holding in real-world situations. Despite this fact it still performs very well in Text Categorization. From the Bayes rule,

\[ P(c_k|d_j) = \frac{P(d_j|c_k)P(c_k)}{P(d_j)} \]

There are two commonly used models for NB classification: Multivariate Bernoulli (MNB) and Multinomial (MNNB). In this work the latter is used, and defining \( N_{ij} \) as the number of times a feature \( x_i \) occurs in a document \( d_j \), the probability of a document given its class is given by [15]

\[ P(d_j|c_k; \hat{\theta}) = \left( \sum_{i=1}^{|V|} N_{ij} \right) ! \prod_{i=1}^{|V|} P(x_i|c_k; \hat{\theta})^{N_{ij}} \]

The probability of \( x_i \) in \( c_k \) is estimated from \( T^* \) and is defined as

\[ \hat{\theta}_{x_i|c_k} = P(x_i|c_k; \hat{\theta}) = \frac{1 + \sum_{n=1}^{P} N_{ij} P(c_k|d_j; \hat{\theta})}{|V| + \sum_{l=1}^{P} \sum_{n=1}^{N_k} N_{ln} P(c_k|d_j; \hat{\theta})} \]

The classification is computed estimating the posterior probability of each class. Since only features that are class dependent are relevant, the contribution of \( \sum_{i=1}^{|V|} N_{ij} \), \( \prod_{i=1}^{|V|} N_{ij} \), and \( P(d_j|\hat{\theta}) \) are discarded. Applying Bayes’ Rule,

\[ P(c_k|d_j; \hat{\theta})=P(c_k|\hat{\theta}) \prod_{i=1}^{|V|} P(x_i|c_k; \hat{\theta})^{N_{ij}} \]

The classification rule for the MNNB is,

\[ \text{class}(d_j) = \arg \max_{c \in C} \left( \log \left( P(c_k|\hat{\theta}) \right) + \sum_{i=1}^{|V|} \log \hat{\theta}_{x_i|c_k} \right) \]

B. Support Vector Machines

SVM were used for solving two-class pattern recognition problems in [16] and introduced in Text Categorization by [17]. It is defined over a vector space where the problem is to find an hypothesis \( h \), or decision surface, that separates the data points into two classes in the most efficient possible way, guaranteeing the lowest error. The true error of \( h \) is the probability that \( h \) will make an error on an unseen or randomly selected test example. Assuming that the linear decision rules are defined by \( h(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) \), where \( \vec{w} \) is a weight vector, orthogonal to the hyperplane and \( b \) a threshold (a parameter that adjusts the margin).

A great advantage of this classifier is that it can use large amounts of input data and feature sets (dismisses the use of FS), therefore being very well adapted for Text Categorization. SVM deals well with high dimension feature vectors both in terms of correctness of the results and efficiency of the categorization algorithm. This is due especially to the fact that SVM has overfitting protection. This leads to the conjecture...
that a good classifier should combine many features. Another argument is that documents vectors are sparse, for each document only a few entries of the feature vector are not zero and SVMs are well suited for problems with dense concepts and sparse instances [17]. LibSVM [18] for python was used in this work, with a linear kernel.

IV. RELATED WORK

Research has been done at different levels of granularity, from document level to word level. At word level features are used to do polarity classification [2]), mostly focused on extracting specific types of information such as adjective-noun relations [19] or nouns that enjoy a dependency relation with a polarity term [20]. At sentence level, [21] identified of sentiment-bearing terms very reliably, [22] presented a system that determined word sentiment and then combined sentiments within a sentence.

The work of [3] is a document level approach that used several supervised ML methods for feature polarity identification within movie reviews, and then classifying the reviews themselves, using accompanying numerical ratings as ground truth. Using the movie corpus of [3] as $T^v$, a small corpus of movie review comments from popular social network Digg was classified according to subjectivity/objectivity and negative/positive attitude [23]. Other approaches select only a subset of the words [3], [24] often by considering only adjectives, [25] identified words describing the features of a product (e.g., The camera takes incredible pictures.) and use them in the classification.

1) Unigrams vs. n-grams: In [3], a feature space composed of unigrams performed better than a bigram or unigrams and bigrams feature set. [19] experimentally showed that trigrams and higher order n-grams failed to show consistent improvement (though bigrams outperformed the solo use of unigrams). There is also the problem of data sparseness of higher order n-grams, as most are rare in any given corpus. Several studies show that a feature space based on n-grams ($n > 1$) is more effective than a unigram based feature space [111], [26]).

A. Cross-Lingual Adaptation

Cross-lingual adaptation is a particular case of domain adaptation, where languages (the domains) have (generally) non-overlapping feature spaces and one needs external knowledge to link features of the source domain to the target domain. ML translation is often used to eliminate the gap between the different language sets.

A pioneer approach to cross-lingual adaptation in Sentiment Analysis was done by [27], where a bilingual lexicon (translated from English) and a manually translated parallel corpus (translated to Romanian) were used to generate the resources required to create target-language training data for developing a statistical classifier. [28] showed that automatic translation is a viable alternative for the construction of resources. ML translation is not perfect and some errors and noise are expected. The error introduced by the automatic translation reduces the performance when compared to the usage of a parallel corpus manually translated [27], however it requires much less resources and time. [29] applied FS to the Expectation-Maximization (EM) algorithm before each iteration (Fig 1).

A co-training approach was proposed by [30] to use with an English labeled corpus and Chinese unlabeled data. Co-training creates two independent views: English view, where the classifier is trained with the English annotated set which will classify the translated unlabeled Chinese set; and the Chinese view, where the classifier is trained with the translated English set which will classify the unlabeled Chinese set. The resulting sets are independent and each document has two prediction values. If the prediction values are conflicting then that document is not classified, otherwise a polarity label is assigned to the unlabeled data. In [31] a new model of co-training was proposed, which concluded that translations are helpful but not enough to assure a good performance.

V. CORPORAS

There are original corpora in three languages: English, Portuguese and Spanish. All corpora were translated to the other two languages using Google Translate. Each corpus is referenced with the original language at bold, and the translated language. For instance, EN-SP, is the original English corpus translated to Spanish.

The rating system differs from site to site so it was necessary to normalize the labels of the collected reviews. For a five star system, reviews higher then 3.5 stars are considered positive and lower than 2 stars (included) as negative, and for a 10 star system, reviews higher then 7 stars are considered positive and lower than 4 stars (included) as negative.

A. IMDB Dataset

The IMDB dataset (EN-EN) is a set of classified movie reviews built by [3]. It now consists of 2000 balanced reviews: 1000 positive and 1000 negative. It is the most popular dataset for Sentiment Analysis research.
B. Spanish Dataset

The Spanish Dataset (SP-SP) was retrieved from the website filmaffinity\(^1\) which gathers movies and TV reviews, in Spanish, from users from all over the world. There are more positive reviews, thus in order to build a balanced corpus they were limited to the same number of negative reviews, to a balanced corpus of 1836 positive and 1836 negative reviews. Following the procedures made in [3] when building the corpus. Reviews and sentences are, in average, shorter than EN-EN.

C. Portuguese Dataset

To perform supervised classification one needs an annotated corpus, which proved to be quite difficult to find in Portuguese. Even an unlabeled corpus referring only to movie reviews is not an easy task due to the lack of resources. The Portuguese corpus is unlabeled, retrieved from the site Cinecartaz\(^2\). The corpus has 1519 reviews of 400 movies. PT-PT has the shortest reviews of all corpora, however the average sentence is bigger than SP-SP.

VI. METHODS

The tasks needs text processing so that the FS process can be as efficient as possible. In this work the natural language structure of text is ignored, there is no semantic analysis, and word order is of no importance (bag-of-words approach). The methods used in this thesis for normalizing text, the proposed novel classification strategies, and the evaluation methodologies are described in this section.

A. Normalizing Text

The text of a document has to be converted into data that a ML algorithm can analyze, e.g., ignoring the difference between The and the. The text is broken down into discrete units (tokenization) and then several operations are applied. The tokenization procedure is described in Table II.

| TABLE II TOKENIZATION ALGORITHM |
|---------------------------------|
| step 1 | Segment the document (coded as UTF-8) into sentences. |
| step 2 | Split sentences into tokens by using white space as delimiters |
| step 3 | Convert all words to lower case (optional) |
| step 4 | Removes all punctuation (optional) |
| step 5 | A list of features is extracted |
| step 6 | Removes stopwords (optional) |

1) **N-grams**: The tokenization of a document will produce features, but identifying multiple token features can produce rarer, highly specific features, such as “New York” or “Madison Square Garden”. This is called a n-gram, where \( n \) determines the number of consecutive tokens used to produce the features.

2) **Performance Measures**: In this work the performance evaluation is made with Accuracy (A) that returns the ratio of correctly labeled inputs per number of documents in the \( T^c \).

3) **k-fold Cross-Validation (CV)**: In \( k \)-fold cross-validation, the original sample is partitioned into \( k \) subsamples. A single subsample is retained as the validation set for testing the model, and the remaining \( k-1 \) subsamples are used as training data. The process is repeated \( k \) times, with each of the \( k \) subsamples used exactly once as the validation data. The \( k \) results from the folds then can be averaged to produce a single estimation of performance.

B. Weighted SVM

The SVM described so far is binary valued, if a feature is present in the document, the corresponding value in the feature vector is 1, otherwise is 0. Other values may be used for the feature vector sent to the kernel. It is proposed to use the scores achieved with FS methods. This method is named WSVM, and referred to as (FS)-WSVM. This method is proposed so that the decision surface is determined according to the relevance of the feature.

C. Add-Polarity and Polarity-SVM

**Add-polarity** is a linear classifier that returns the label of a document, according to the sum of the features polarities, measured with the \( \text{pol}(w) \) function,

\[
\text{pol}(w) = \begin{cases} 
0, & \text{if } \sum w \leq \text{DF}(w, \text{pos}) - \text{DF}(w, \text{neg}) > 0 \\
0, & \text{if } \sum w \leq \text{DF}(w, \text{pos}) - \text{DF}(w, \text{neg}) = 0 \\
1, & \text{if } \sum w \leq \text{DF}(w, \text{pos}) - \text{DF}(w, \text{neg}) < 0 
\end{cases}
\]

If a document is being classified at sentence and word level then the polarity of the document is the sum of the polarity of the sentences, that are in turn classified with the polarity of the features contained therein,

\[
\text{pol}(\text{doc}) = \begin{cases} 
0, & \text{if } \sum w \leq \text{pol}(\text{sent}) > 0 \\
0, & \text{if } \sum w \leq \text{pol}(\text{sent}) = 0 \\
1, & \text{if } \sum w \leq \text{pol}(\text{sent}) < 0 
\end{cases}
\]

**Polarity-SVM** (PSVM) is a SVM whose feature vector labels are the polarity of the features, thus taking values in \( \{-1, 0, +1\} \).

D. Granularity

A document is a group of words mixed together, however, it may also be assumed as a group of sentences, which are a group of words. The deconstruction of a review at lower levels, may be insightful as to how much sentiment can be captured in smaller levels of classification.

1) **Document Classification at Word Level**: This method is normally called word level (Fig. 2). The classifiers used are Add-Polarity and PSVM, since these classifiers are based on features’ polarities.

\(^1\)www.filmaffinity.com
\(^2\)http://cinecartaz.publico.pt/
2) **Sentence Classification:** Sentence classification is mostly used on its own in subjectivity classification, where one wants to filter all the non-subjective sentences, and then classify a subjective document according to its sentiment. When reading a movie review, sometimes not all text refers to an actual review, but it is rather a reference to other reviews or even a summary of the plot. In the context of Sentiment Analysis those remarks are not relevant. Classifying a sentence as either positive or negative with training categories based on the category of the reviews it was taken from is not very effective. A review may be positive and have some negative remarks. This will bias the data and confuse the classifier.

3) **Document Classification at Sentence Level:** Sentence classification may be used to classify documents (Fig. 3). This test elicits the verification of whether sentence classification is coherent with the reviews polarity from which it was retrieved. The contribution of sentences to the overall document classification will also permit, for instance, the identification of sentences that are not classified with the same polarity of the review providing a means to identify irony.

**E. Poly-Lingual Training**

This method builds a $\mathcal{T}^r$ with reviews from all corpora to train a classifier. The feature space is composed of the union of all vocabularies. Since all translated corpora were constructed using Google Translate, and the unknown words are not translated, the translated corpora will have words pertaining to the original corpus. Also, it is quite natural that English words also appear in the Spanish corpus (the Portuguese corpus will not be tested here because it is unlabeled), since those reviews are bound to mention movie titles, or expressions from popular American movies (Fig. 4).

**F. Cross-Lingual Training**

This method makes use of the translated/original corpora pair for each language. There are two approaches, one in which the classifier is trained with a corpus in its original language and used to test a translated corpus, and vice-versa (Fig. 5). This approach will allow to combine the availability of the English corpora and the ML translation services to classify corpora that lacks huge amounts of labeled data.

**VII. RESULTS**

Punctuation, stopwords and lowercased words were used in all experiments, with 10-fold CV, and unless stated otherwise without FS. Only features that appear a minimum number of times in the $\mathcal{T}^r$ were kept. Performances were tested for several thresholds ranging from 0 to 7 and the minimum of 2 yielded the best average performance and was thus chosen as standard for this work.

**A. Classification and Feature Selection Methods**

MNNB and SVM performances regarding the number of features used and FS method used are shown in Fig. 6(a)-6(b). The FS methods that consistently performed better with both classifiers were DF, GiniB$_{max}$, CET and CHI. GUM$_{max}$ performed very well with the SVM, but it performed poorly with MNNB. The classification performances obtained can be seen in Table III. SVM outperformed MNNB in the EN-EN corpus (reaching a 5.0% pp difference). However, it was outperformed in SP-SP. The best performance of all corpora was achieved using unigrams, bigrams and trigrams. Unigrams alone perform better than bigrams and much better than trigrams, with the SVM. Using MNNB this difference was more noticeable with the SP-SP corpus. Using only bigrams and trigrams performed better than bigrams and trigrams with
### Table III
Document Classification at several levels

| Corpus   | Granularity | Classifier | Unigrams | Bigrams | Trigrams | Unigrams and Bigrams | Bigrams and Trigrams | Unigrams, Bigrams and Trigrams |
|----------|-------------|------------|----------|---------|----------|----------------------|----------------------|-------------------------------|
| EN-EN    | Document Level | MNNB        | 0.8130   | 0.8140  | 0.6690   | 0.8250               | 0.8090               | 0.8250                        |
|          | Document Level | SVM         | 0.8580   | 0.8110  | 0.6230   | 0.8670               | 0.8080               | 0.8710                        |
|          | Word Level    | Unigrams    | 0.4600   | 0.5900  | 0.3300   | 0.6600               | 0.6400               | 0.7600                        |
|          |              | Bigrams     | 0.7600   | 0.7400  | 0.6400   | 0.7600               | 0.7400               | 0.7600                        |
|          |              | Trigrams    | 0.6690   | 0.6230  | 0.3300   | 0.6300               | 0.6000               | 0.6000                        |
|          | Sentence Level | PSVM        | 0.5291   | 0.5502  | 0.5176   | 0.5608               | 0.5506               | 0.5615                        |
| SP-SP    | Document Level | MNNB        | 0.9280   | 0.8760  | 0.5980   | 0.9350               | 0.9283               | 0.9350                        |
|          | Document Level | SVM         | 0.9010   | 0.8160  | 0.5840   | 0.9140               | 0.8160               | 0.9150                        |
|          | Word Level    | Unigrams    | 0.5500   | 0.7100  | 0.6400   | 0.7400               | 0.7500               | 0.7600                        |
|          |              | Bigrams     | 0.7200   | 0.8300  | 0.7700   | 0.8600               | 0.8600               | 0.8800                        |
|          |              | Trigrams    | 0.8574   | 0.6153  | 0.5804   | 0.6352               | 0.6292               | 0.6487                        |
|          | Sentence Level | PSVM        | 0.8061   | 0.4291  | 0.2500   | 0.7888               | 0.4412               | 0.7981                        |
|          |              | Bigrams     | 0.8128   | 0.4011  | 0.2500   | 0.8195               | 0.8091               | 0.8422                        |
|          |              | Trigrams    | 0.9222   | 0.9208  | 0.8724   | 0.9307               | 0.9160               | 0.9283                        |

The table above shows the classification results for different classifiers at various levels of granularity. The performance is measured using accuracy values.

### B. Weighted SVM

For each labeled corpus and FS method studied, the results for a feature space composed of unigrams, bigrams and trigrams can be seen in Table IV. Even though MI did not have a good performance when comparing FS methods, it was the best FS to use as weight with the WSVM. Also, the OR generally performed very well, almost as well as the SVM, but once more this difference is less noticeable with the MNNB. The translated corpora and the original corpus behaved similarly, with a small performance decrease.

#### Table IV
Weighted SVM Using Trigrams, Bigrams and Unigrams

|              | EN-EN | EN-PT | EN-SP | SP-SP | SP-SP | SP-PT | SP-EN |
|--------------|-------|-------|-------|-------|-------|-------|-------|
| MI\(_{max}\) | 0.8795| 0.8790| 0.8665| 0.9291| 0.9232| 0.9255|       |
| MI\(_{avg}\) | 0.8695| 0.8625| 0.8490| 0.9176| 0.9224| 0.9240|       |
C. Granularity

1) Word Level Classification: As one can see from Table III, results at word level are far from the ones achieved at document level. Performance with the SP-SP corpus is still higher than with EN-EN, and a feature space composed of unigrams, bigrams and trigrams always leads to the best performance. Using only unigrams or trigrams lead to the worst performances, however using only bigrams generally achieves much higher performance. When using bigrams and trigrams the performance is always higher or equal to bigrams and unigrams. Add-Polarity is a very simple classifier system and, as expected, its performance is sometimes below 50.0%. PSVM outperformed Add-Polarity in all corpora.

2) Sentence Classification: There is hardly any difference between feature sets composed of n-grams combinations, mainly due to the lack of features per sentence, and a unigrams feature space behaved the best. MNNB was the classifier with the best performance in all corpora.

Table V

| Classifier | EN-EN | SP-SP |
|------------|-------|-------|
| PSVM       | 0.5616 | 0.6488 |
| SVM        | 0.6022 | 0.6900 |
| WSVM       | 0.5959 | 0.6573 |
| MNNB       | 0.6554 | 0.7516 |

3) Document Classification at Sentence Level: As shown in Table III MNNB outperformed all other classifiers at sentence level, which is coherent with the sentence classification results. MNNB classifier at sentence level outperformed the document level classification, except for the SP-SP corpora. These results show that sentence contribution is very effective if properly classified, increasing performance of the classifier by 2.0% pp (on average). If sentences are so informative about document polarity, then sentences that are mislabeled may in fact be very informative about irony or objective information. Also WSVM generally outperformed SVM and behaved poorly with trigrams, which is once more expected due to the few non zero terms of the feature vectors a sentence $T^r$. Add-polarity behaved very poorly, and PSVM not behaving so bad, was far from the performance of the other classifiers. Once more, a $T^r$ composed of bigrams and unigrams or trigrams (or both) achieved the best performance, coherently with the previous results that hinted that bigrams contribute the most to a better performance.

D. Poly-Lingual Training

Three different classifiers were used: MNNB, SVM and (MI$_{max}$)-WSVM. The FS method used was GiniB$_{max}$. The $T^r$ contains documents from each of the corpora in the $T^r$. Performance results can be seen in Table VI. The feature space composed of unigrams, bigrams and trigrams had the best performance and was the one shown.

Poly-Lingual proved to be very effective, especially when using (MI$_{max}$)-WSVM. Using only the 10.0% most relevant features of the translated feature space was not a good option, being outperformed in all experiments. The difference of using translated and original corpora opposed to using only original corpora is barely noticed, with a small improvement with the MNNB. As already referenced, SVM deals well with a large dimensional space. The performance of the (MI$_{max}$)-WSVM is the best in this study, outperforming all other evaluations.

E. Cross-Lingual Training

It is shown in Table, VII both combinations of translated/original performance, and it was also tested a third point of view, using both Portuguese translated corpora (EN-PT and SP-PT). Three classifiers were used: MNNB, SVM and (MI$_{max}$)-WSVM. Trying to reduce the noise introduced by the translator, only the 10% most relevant features (GiniB$_{max}$) were used as features from the translated corpus feature set.

Table VII

| Train | Test | MNNB | SVM | WSVM |
|-------|------|------|-----|------|
| EN-SP | SP-SP | 0.8568 | 0.8066 | 0.8601 |
| EN-EN | EN-EN | 0.7375 | 0.7805 | 0.7935 |
| SP-PT | PT-PT | 0.7440 | 0.8215 | 0.8745 |

Without FS, MNNB was outperformed by SVM and WSVM, however reducing the number of translated features yielded better performances, especially when training with the translated corpora. WSVM outperformed all other classifiers when using all features available, however in the reduced feature space it was the one outperformed most of the times, sometimes with a decrease of more than 10.0% pp comparing with the previous result. WSVM performs best when it uses all the feature space.

In English, training with EN-EN leads to better performances and in Spanish training with SP-EN gave the best performance. Whenever EN-EN was used the best performances were achieved, which may be due to the characteristics of the corpus (longer and selected reviews). These performances are actually high, taking into account that when using EN-EN as training set and SP-EN as a test set, the performance is almost as good as one achieved with standard classification for the EN-EN with the WSVM. This suggests that the EN-EN generalizes well and is perhaps a good way to use available English data to improve classification in other languages.

The translated Portuguese corpora achieved very good results using FS. FS was used here to eliminate noise caused by automatic translation, and using the 10% most relevant features MNNB had the best performance, which was somewhat expected, since feature vectors are of a lower dimension and it was already shown that MNNB works better than SVM with
low dimension feature vectors. Results achieve a surprising performance of 96.5% for EN-PT / SP-PT and 94.1% for SP-PT / EN-PT, which are very high performance values.

F. Semi-Supervised Classification

The main idea is to improve classification performance with information from the unlabeled data. Labeled data will always be necessary for the $T^e$. Both an English-Portuguese and Spanish-Portuguese experiment was done and a Portuguese $T^e$ manually classified from the same source of the unlabeled data was used to validate the classifications.

1) Expectation-Maximization: The first approach is the basic algorithm already described (Fig. 1). The target language is Portuguese and there are two experiments, one with Spanish as the source language and another with English (NB classifier). When FS is applied (GiniB), only the 10% most relevant features are used. The performance for both approaches is shown in Table VIII.

![Table VIII EM results](image)

The English view needed one more iteration, with and without FS, and both cases converged to the same accuracy value, which indicates that both English and Spanish converged to the same results. The final $T^e$ has a negative impact in the system, achieving a decrease of about 20.0% pp. Using FS in the $T^e$ worked very poorly. Note that the problem with this algorithm in this particular task is that the classifier ends with almost no negative reviews, since in each iteration negative reviews are reconsidered as positive. Perhaps a corpus with longer reviews could prove to be more effective.

2) Co-Training: Following [30], the Co-Training approach is tested from the English-Portuguese and Spanish-Portuguese point of view. There is an independent $T^e$, composed of 200 reviews (100 negative and 100 positive), manually annotated and randomly selected, that will remain unaltered. The tests done are: SVM$_{PTEN}$ SVM$_{PTSP}$ (Cross-lingual using the independent $T^e$ as $T^e$ and the EN-PT and SP-PT corpora as $T^e$); SVM$_{EN}$ SVM$_{SP}$ (Cross-lingual using the EN-EN and SP-SP as the $T^e$ and the independent $T^e$ as $T^e$); SVM$_{ENPT1}$ SVM$_{SPPT1}$ (inductive SVM with English (or Spanish) and Portuguese features for classification in the two views); SVM$_{ENPT2}$ SVM$_{SPPT2}$ (combines the results of SVM$_{EN}$ (SVM$_{SP}$) and SVM$_{PTEN}$ (SVM$_{SPPT}$) by averaging the prediction values); and Co-Train$_{EN}$, Co-Train$_{SP}$ (Co-training technique).

The best pair $p$ and $n$ was achieved with $p=n=5$ and a maximum number of 100 iterations. Only the SVM classifier was used in this experiment. So that this experiment is easily compared with the results achieved by [30] the same baselines were compared. The results are shown in Table IX.

![Table IX Co-Training](image)

The Co-training technique was not an improvement over the baseline proposed. It did not outperform cross lingual performance and both Spanish and English view behaved very poorly. Both in SVM$_{ENPT1}$ and Co-Train$_{EN}$ there is a comparison between the output of the classifier, and the fact that Portuguese and Spanish are so much more similar than Spanish may be an indicator as to why in English the performances decreased so drastically comparing with the cross lingual adaptation and in Spanish there was only a slightly decrease. In the English point of view, there was a tendency of the classifier to classify reviews as positive, indicating a bias towards positive reviews.

VIII. CONCLUSION

The FS methods that performed best were DE, GiniB$_{max}$, CET and CHI. SVM is a much faster algorithm than MNB, it is less prone to overfitting and can scale up to considerably large dimensions, so it is expected that for FS to have a lower impact. However, the best FS methods are common to both classifiers. Classifiers perform better without FS.

The proposed W SVM outperforms SVM in all tests, as long as that there is no FS, and when the corpus is composed of longer reviews (e.g., EN-EN). MNB works better with shorter review corpus (e.g., SP-SP), or when classifying sentences). Which leads to think that when constructing a subjective/objective filter or an irony identifier, MNBN may be preferable to SVM.

Using mixed a n-grams feature space proved to be the best option, specially using all n-grams, which generally outperformed all other combinations.

Classification at different levels of granularity revealed that document classification at document level is the most effective way for both corpus, if using a feature space composed of unigrams, bigrams and trigrams. In all the other feature space combinations sentence level performs better. Word level has the worst performance and the two classifiers created for this task (Add-Polarity and PSVM) are not good classifiers.

WSVM was the best classifier with all experiments of the Poly-Lingual method. The accuracy of the classifier is higher than the one achieved with the EN-EN corpus, but worse than the SP-SP, and closer to the one achieved with the SP-SP corpus, which is expected since it has more reviews. It suggests that since both corpus are independent, the classification is also made independently. At the Cross-Lingual method WSVM
was the best classifier with no FS, and MNNB with FS. Using the EN-EN corpus as $T^*$, without FS, achieved a good performance, specially due to the fact the translation is automatic and without supervision, shedding some light about how to use English resources automatically. Results using the Portuguese translated corpora is also very interesting. Since both were automatically translated, results may be biased with the translation outcome.

Semi-supervised techniques were proposed to label Portuguese data, however it was not successful in achieving good performances, actually, decreasing accuracy values as unlabeled reviews were being added as input after each iteration. This results indicate that using the chosen techniques, unlabeled corpora does not provide provide correct information to the classification system. Portuguese and Spanish co-training, however, worked much better than Portuguese and English, which may be due to the similarity between languages. This proximity between languages is important, because lacking a labeled corpus in Portuguese, the Spanish corpus is a hint on how a Portuguese corpus may behave. In general, unlabeled reviews were classified as positive, which may in fact be true, because negative reviews are very scarce to find.

The construction of a classification system that is immune to what language is used in the data is very appealing. Releasing human work from the classification process is very seducing, however the lack of resources still do not allow for a framework of this kind to be built. The shown distinction of behavior between classifiers allows for a better choice of classification system according the specification of the problem.

REFERENCES

[1] F. Sebastiani, “Machine learning in automated text categorization,” ACM computing surveys (CSUR), vol. 34, no. 1, pp. 1–47, 2002.
[2] P. D. Turney, “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews,” in ACL ’02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 417–424.
[3] B. Pang, L. Lee, and S. V. H. A. N. Y. A. D. I. R. L. S. M. T. E. R. I. O. N. A. S. S. T. I. C. E. S. T. Y. N. G. S. Y. S. T. E. E. M. S. K. N. O. W. L. D. E. E. N. D. D. C. Y. S. (FSKD), 2010 Seventh International Conference on, vol. 6. IEEE, 2010, pp. 2513–2517.
[4] Y. Chen and O. Wu, “Semi-automated feature selection for web text filtering,” in Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on, vol. 6. IEEE, 2010, pp. 2513–2517.
[5] Y. Yang, “An evaluation of statistical approaches to text categorization,” Information retrieval, vol. 1, no. 1, pp. 69–90, 1999.
[6] K. Gao, T. Khoshgoftaar, H. Wang, and N. Seliya, “Choosing software metrics for defect prediction: an investigation on feature selection techniques,” Software: Practice and Experience, vol. 41, no. 5, pp. 579–606, 2011.
[7] D. Mladenic and M. Grobelnik, “Feature selection for unbalanced class distribution and naive bayes,” in MACHINE LEARNING—INTERNATIONAL WORKSHOP THEN CONFERENCE—MORGAN KAUFMANN PUBLISHERS, INC., 1999, pp. 258–267.
[8] Z. Zheng, X. Wu, and R. Srivastava, “Feature selection for text categorization on imbalanced data,” ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 80–89, 2004.
[9] H. Park, S. Kwon, and H. Kwon, “Complete gini-index text (gini) feature-selection algorithm for text classification,” in Software Engineering and Data Mining (SEDIM), 2010 2nd International Conference on. IEEE, 2010, pp. 366–371.
[10] K. Ngam, A. McCallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using em,” Machine learning, vol. 39, no. 2, pp. 103–134, 2000.
[11] V. Ng and S. Dasgupta, “Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews,” in Proceedings of the COLING-WC on …, 2006.
[12] S. Li, B. Xia, C. Zong, and C. Huang, “A framework of feature selection methods for text categorization,” in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. Association for Computational Linguistics, 2009, pp. 692–700.
[13] D. Mladenic, “Machine learning on non-homogeneous,” Distributed Text Data Doctoral Dissertation, University of Ljubljana, 1998.
[14] G. Uchyigit and K. Clark, “A new feature selection method for text classification,” in International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, no. 2, p. 423, 2007.
[15] A. McCallum and K. Nigam, “A comparison of event models for naive bayes text classification,” AAAI-98 workshop on learning for text …, 1998.
[16] V. Vapnik, “The nature of statistical learning.” 1995.
[17] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” Machine Learning: ECML-98, pp. 137–142, 1998.
[18] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” ACM Transactions on Intelligent Systems and Technology, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[19] K. Dave, S. Lawrence, and D. Pennock, “Mining the peanut gallery: opinion extraction and sentiment classification of product reviews,” in Proceedings of the 12th international conference on World Wide Web. ACM, 2003, pp. 519–528.
[20] A. Popescu and O. Etzioni, “Extracting Product Features and Opinions from Reviews,” in Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HL’05, 2005, pp. 339–346.
[21] P. Turney and M. Littman, “Measuring praise and criticism: Inference of semantic orientation from association,” in ACM Transactions on Information Systems (TOIS), 2003.
[22] S. Kim and E. Hovy, “Determining the sentiment of opinions,” in Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004, pp. 1367–8.
[23] K. Yessenov and S. Misailovic, “Sentiment analysis of movie review comments,” 2009.
[24] J. Wiebe, “Learning subjective adjectives from corpora,” in Proceedings of the National Conference on Artificial Intelligence. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2000, pp. 735–741.
[25] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004, pp. 168–177.
[26] A. Pak and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining,” in Proceedings of LREC, vol. 2010, 2010.
[27] R. Mihalcea, C. Banea, and J. Wiebe, “Learning multilingual subjective language via cross-lingual projections,” in ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, vol. 45, no. 1, 2007, p. 976.
[28] C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan, “Multilingual subjectivity analysis using machine translation,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2008, pp. 127–135.
[29] L. Rigatini, M. Maggini, and B. Liu, “An em based training algorithm for cross-language text categorization,” in Web Intelligence. 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on. IEEE, 2005, pp. 529–535.
[30] X. Wan, “Co-training for cross-lingual sentiment classification,” in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. Association for Computational Linguistics, 2009, pp. 235–243.
[31] B. Wei and C. Pal, “Cross lingual adaptation: an experiment on sentiment classifications,” in Proceedings of the 20th conference on Empirical methods in Natural Language Processing. Association for Computational Linguistics, 2010, pp. 258–262.