Making Predictions with Textual Contents

Experiments with Portuguese Documents from Three Distinct Domains

Indira Mascarenhas Brito Bruno Martins
Instituto Superior Técnico, INESC-ID
{indira.brito,bruno.g.martins}@tecnico.ulisboa.pt

Abstract

Forecasting real-world quantities, with basis on information from textual descriptions, has recently attracted significant interest as a research problem, although previous studies have focused on applications involving only the English language. This paper presents an experimental study on the subject of making predictions with textual contents written in Portuguese, using documents from three distinct domains. We specifically report on experiments using different types of regression models, using state-of-the-art feature weighting schemes, and using features derived from cluster-based word representations. Our experiments show that prediction models using the textual information achieve better results than simple baselines such as the average value, and that richer document representations (i.e., using Brown clusters and the Delta-TF-IDF feature weighting scheme) may result in slight performance improvements.

Keywords

Text-Driven Forecasting, Learning Regression Models, Word Clustering, Feature Engineering

1 Introduction

Text-driven forecasting has recently attracted a significant interest within the Information Extraction (IE), Information Retrieval (IR), Machine Learning (ML), and Natural Language Processing (NLP) international communities [Smith, 2010, Radinsky, 2012]. Well-known examples of previous studies include using textual contents for making predictions about stock or market behavior [Luo et al., 2013, Lerman et al., 2008, Tirunillai and Tellis, 2012, Schumaker and Chen, 2009, Bollen et al., 2011], sports betting market results [Hong and Skiena, 2010], product and service sales patterns [Chahuneau et al., 2012, Joshi et al., 2010], government elections, legislative activities and general political leans [Yano et al., 2012, Dahllöf, 2012], or general public opinion polls [Mitchell et al., 2013, O'Connory et al., 2010, Schwartz et al., 2013].

This paper presents an experimental study on the subject of making predictions with textual contents written in Portuguese, using documents from three distinct domains, namely (i) descriptions for hotels in Portugal collected from a well-known Web portal, associated with average room prices in the high and low seasons for tourists, (ii) descriptions for restaurants and the corresponding menus, also collected from the same Web portal, associated with the average meal prices, and (iii) movie reviews collected from a specialized Web site, together with the corresponding box-office results for the first week of exhibition, as available from Instituto do Cinema e do Audiovisual. Our study focused on the usage of machine learning methods from the current state-of-the-art (e.g., Random Forest regression, or linear regression with Elastic Net regularization), as implemented in an open source Python machine learning library named scikit-learn[1]. Besides the issue of Portuguese contents, our study also introduces some technical novelties in relation to most previous work in the area, namely by (i) experimenting with state-of-the-art IR feature weighting schemes such as Delta-TF-IDF or Delta-BM25, and by (ii) experimenting with features derived from cluster-based word representation (i.e., using Brown clusters).

The rest of this paper is organized as follows: Section 2 presents previews work related to text-driven forecasting. Section 3 details the contributions of the paper, presenting the regression techniques that were considered, as well as the approaches taken for representing the textual contents as feature vectors. Section 4 presents the experimental evaluation, describing the datasets from the three different domains, and discussing the obtained results. Finally, Section 5 presents the main conclusions and points possible directions for future work.

[1] scikit-learn.org

2 Related Work

Taking inspiration from recent research on sentiment analysis [Pang and Lee, 2008], in which machine learning techniques are used to interpret text based on the subjective attitude of the author, Noah Smith and his colleagues have addressed several related text-mining tasks, where textual documents are interpreted to predict some extrinsic, real-valued outcome of interest, that can be observed in non-text data. This group is perhaps the one with the highest level of activity in this specific research problem. A relatively recent white paper, summarizing the work that these researchers have been developing, has been published online by Smith [2010]. Specific examples for the text-driven forecasting tasks that these authors have addressed include:

1. The interpretation of an annual financial report published by a company to its shareholders, in order to try predicting the risk incurred by investing in that same company, in the coming year [Kogan et al., 2009].
2. The interpretation of a movie critic's textual review of a film, to try predicting the film's box-office success [Joshi et al., 2010].
3. Interpreting political blog posts, to try predicting the response that they will gather from the readers [Yano and Smith, 2010].
4. The interpretation of a day's microblog feeds, in order to try predicting the public's opinion about a particular issue [Yano et al., 2009, Yano and Smith, 2010].
5. The interpretation of food writings, as given on restaurant descriptions, on restaurant menus, and on customer reviews, to try predicting average meal price and customer rantings [Chahuneau et al., 2012].

In all of the above cases, one aspect of the text's meaning is observable from objective real-world data, although perhaps not immediately at the time the text is published (i.e., respectively, we observe the return volatility, the gross revenue, the user comments, measurements from traditional opinion polls, and average meal prices, in the five problems that were previously enumerated). Smith [2010] proposed a generic approach to text-driven forecasting, based on fitting regression models that leverage on features derived from the text, which are generally noisy and sparse. He argued that text-driven forecasting, as a research problem, can be addressed through learning-based methodologies that are neutral to different theories of language. At the same time, an attractive property of this line of research is that the evaluation is objective, inexpensive, and theory-neutral [Smith, 2010].

On what regards movie reviews and gross revenues, and summarizing the previous research presented by Joshi et al. [2010], Smith mentioned that before a movie premiere, critics attend advance viewings and publish textual reviews about them. The authors considered making predictions about the box-office results of movies with basis on the text from these reviews, right after they are produced by expert critics. The authors considered 1,351 movies released between January 2005 and June 2009. For each movie, two kinds of data were obtained:

1. Descriptive metadata was gathered from *Metacritic*[2], which includes the name, production house, genre(s), scriptwriter(s), director(s), primary actors, and the country of origin, among other information. Metadata from a website called *The Numbers*[3] was also gathered, containing information about the production budget, opening weekend gross revenues, and number of screens on which the movie played in that weekend.
2. Reviews were extracted from the six review Web sites that appeared most frequently at *Metacritic*, only considering the reviews made before the release date of the movie.

Smith described the application of linear regression modeling with state-of-the-art Elastic Net regularization [Zou and Hastie, 2005, Fridman et al., 2008]. The model was trained on 988 examples released from 2005-2007, and it was evaluated by forecasting box-office revenue for each film released between September 2008 and June 2009 (i.e., a total of 180 movies). The authors calculated the Mean Absolute Error (MAE) on the test set, analysing the difference between the estimated revenue generated by a movie during its release weekend, and the actual gross earnings, per screen. Models that use the text alone (MAE of \$6,729) or in addition to metadata (MAE of \$6,725) were better than models using only the metadata (MAE of \$7,313). Text reduces the error by 8% compared to metadata, and by 5% against the strong baseline of predicting box-office results with basis on the median value at movies from the training data.

Regarding the prediction of risk from financial reports, and with basis on previous research developed by Kogan et al. [2009], Smith said that

[2] www.metacritic.com

[3] www.the-numbers.com

predicting returns and profit is indeed a difficult problem, given the risk (i.e., the volatility or standard deviation of returns, over a period of time). In these experiments, the authors considered annual reports known as *Form 10K*, seeking to predict volatility (i.e., an indicator to risk) in the year following a report's publication.

A total of 26,806 *Form 10K* reports were collected, consisting of a quarter billion words, from 1996-2006. Financial data were also used to calculate the volatility for the firm that published each report in two periods, namely in the twelve months $V^{(-12)}$ before the reports, and in the twelve months $V^{(+12)}$ after. The aim was to predict the months after the reports, because volatility shows strong autocorrelation. The authors used a linear regression model, based on support vector regression [Drucker et al., 1997], to predict $V^{(+12)}$ from word and bigram frequencies in Section 7 of the *Form 10K* reports, including $V^{(-12)}$ as a optional feature. Smith discussed one set of experimental results where the volatility for 3,612 firms was predicted, following their 2003 *Form 10K* reports. The text-only model outperformed the models based on a baseline that corresponds to the volatility in $V^{(-12)}$, in terms of the Mean Squared Error. Having models that use the two types of data works even better.

In what concerns the interpretation of political blogs, Smith reported the main results of the previous research made by Yano and Smith [2010] and by Yano et al. [2009]. The authors created a dataset containing 79,030 blog posts extracted from five American political blogs in 2007 and 2008. For each post, the readers can leave their comments. Smith reported on studies in which the authors tried to predict the individual [Yano et al., 2009] and aggregate [Yano and Smith, 2010] behavior of blog readers, using hidden-variable models based on Dirichlet allocation. These models produce not just a forecast, but clusters that tend to be topically coherent and that can be quantitatively linked to the predictions. The authors also used the CommentLDA model [Yano et al., 2009] to predict the five most likely comments per new post. The model achieved a precision of 27.5%, which is a good result when compared to a Naïve Bayes bag-of-words baseline that achieved 25.1%. The model also discovered topics relating to *religion*, *domestic policy*, and the *Iraq war*, among others. For comments with a number of words that is higher than the average, the authors built a model combining CommentLDA with Poisson regression [Armstrong and Collopy, 1987]. The precision of this model dropped slightly when compared to a Naïve Bayes bag-of-words model (72.5% to 70.2%), but recall significantly increased from 41.7% to 68.8%.

In his survey paper, Smith also summarized the research by O'Connory et al. [2010], connecting measures of public opinion collected from polls, with population-level sentiment measured from text. For this experiment, the authors used two kinds of data, namely text data from Twitter, and public opinion survey data from multiple polling organizations. The messages on Twitter are short, averaging 11 words per message. A total of 1 billion Twitter messages, posted over the years of 2008 and 2009, were collected by querying the Twitter API. For the *ground-truth* public opinion, several measures of consumer confidence and political opinion were considered. The consumer confidence refers to how optimistic the public feels, regarding the economy and their personal finances. The main goal was assessing the population's aggregate opinion on a topic, and the results showed that a simple aggregate score, based on positive and negative sentiment word frequencies, closely tracks a time series of tremendous interest to investors, i.e., the consumer confidence. The tweets that mentioned the word *economy* derived the score, and one of the specific indexes that the authors tried to predict was Gallup's[4] economy confidence index.

More recently, Chahuneau et al. [2012] explored the interactions in language use that occur between restaurant menu prices, menu descriptions, and sentiments expressed in user reviews, from data extracted from *Allmenus.com*[5]. From this Web site, the authors gathered menus for restaurants in seven North American cities, namely *Boston*, *Chicago*, *San Francisco*, *Los Angels*, *New York*, *Washington D.C.*, and *Philadelphia*. Each menu contains a list of item names, with optional textual descriptions and prices. Additional metadata (e.g., price range, location, and ambiance) and user reviews (i.e., textual descriptions associated to rantings in a 5-star scale), for most of the restaurants, were collected from a service named *Yelp*[6].

The authors considered diverse forecasting tasks, such as predicting individual item prices, predicting price range for each restaurant, and jointly predicting median price and sentiment. For the first two tasks, the authors used linear regression, and for the third task, they used logistic regression models, all with l_1 regularization

[4] www.gallup.com/poll/122840/gallup-daily-economic-indexes.aspx

[5] www.allmenus.com

[6] www.yelp.com

when sparsity is desirable. For the evaluation, they used metrics like the Mean Absolute Error (MAE) or the Mean Relative Error (MRE).

When predicting the price of each individual item on a menu, Chahuneau et al. [2012] used the logarithm of the price as the output value, because the price distribution is more symmetric in the log domain. The authors evaluated several baselines that make independent predictions for each distinct item name. Two simple baselines use the mean and the median of the price, in the training set and given the item name. A third baseline used a l_1-regularized linear regression model, that was trained with multiple binary features, one for each item name in the training data. They performed a simple normalization of the item names for all the baselines, due to the large variation of menu item names in the dataset (i.e., there were more than 400,000 distinct names). The normalization consists in removing stop words compiled from the most frequent words in the item names, and ordering the words in each item name lexicographically. This normalization reduced the unique item name count by 40%.

The authors also used several feature-rich models based on regularized regression, considering (i) binary features for each restaurant metadata property, (ii) n-grams in menu item names, with n-grams corresponding to sequences of n tokens (i.e., with $n \in \{1, 2, 3\}$) from a given sentence, (iii) n-grams in the menu item descriptions, and (iv) n-grams from mentions of menu items in the corresponding reviews. When using the complete set of features, the authors report on a final reduction of 50 cents in the MAE metric, and of nearly 10% in MRE, a good result when compared with the baselines.

For the task of predicting the price range, the target values were integers from 1 to 4 that denote the price of a typical meal from the restaurant. For the evaluation of this specific task, the authors rounded the predicted values to integers, and used the Mean Absolute Error (MAE) and the Accuracy evaluation metrics. They achieved a small improvement when comparing their linear regression model with an ordinal regression model (i.e., a regression model that assigns, to each instance, a ranking value between one and four, and that takes the ordering of the target values into consideration [McCullagh, 1980]), measuring 77.32% of Accuracy against 77.15%, for models with metadata features. They also used features from the complete text of the reviews, besides the features used for the task of predicting the individual menu item prices. By combining metadata and review features, the measured Accuracy exceeds 80%.

For the task of analysing the sentiments expressed in review texts, the authors trained a logistic regression model, predicting the polarity for each review. The polarity of a review was determined by the corresponding star rating, i.e., checking if it was above or below the average rating. The obtained Accuracy was 87%.

Finally, Chahuneau et al. [2012] considered the task of predicting aggregate price and sentiment for a restaurant. To do this, they tried to model, at the same time, the review polarity $\overline{r}$ and the item price $\overline{p}$. They calculated, for each restaurant in the dataset, the median item price and the median star rating. A plane ($\overline{r}$,$\overline{p}$) was divided into four sections, with the average of these two values in the dataset as the origin coordinates, namely $8.69 for $\overline{p}$ and 3.55 stars for $\overline{r}$. This division allowed the authors to train a 4-class logistic regression model, using the features extracted from the reviews for each restaurant. The obtained Accuracy was, in this case, of 65%.

3 Making Predictions With Text

In our study, similarly to Noah Smith and his colleges, we approached the problem of making predictions from textual contents as a regression task. Each document is modeled as a vector of characteristics in a given vector space, in which the dimensionality corresponds to the number of different features. This representation is associated with a well-known model for processing and representing documents in the area of Information Retrieval, commonly referred to as the vector space model. Formally, we have that each document is represented as a feature vector $\overrightarrow{d_j} =< w_{1,j}, w_{2,j}, ..., w_{k,j} >$, where k is the number of features, and where $w_{i,j}$ corresponds to a weight that reflects the importance of feature i for describing the contents of document j. The features are essentially the words that occur in the document collection, but in some of our experiments we also tried other features, such as metadata referring to the geographic location (i.e., the administrative districts) associated to the instances, the type of restaurants, or word clusters associated to the textual tokens occurring in the corresponding document.

3.1 Word Clustering

Word clustering essentially aims to address the problem of data sparsity, by providing a lower-dimensional representation for the words in a document collection. In this work, we used the word clustering algorithm proposed by Brown et al. [1992], which induces generalized representations of individual words. Brown's algorithm is essentially a process of hierarchical clustering that groups words with common characteristics, in order to maximize the mutual information of bi-grams. The input for the algorithm is a textual corpus, which can be seen as a sequence of N words $w_1, \cdots, w_N$. The output is a binary tree, in which the leaves of the tree are the words. The clustering process is related to a language model based on bi-grams and classes:

$$\mathrm{P}(w_1^N|C) = \prod_{i=1}^{N} \mathrm{P}\left(\mathrm{C}(w_i)|\mathrm{C}(w_{i-1})\right) \times \mathrm{P}\left(w_i|\mathrm{C}(w_i)\right)$$

In the formula, $\mathrm{P}(c|c')$ corresponds to the transition probability for the class c given its predecessor class c', and $\mathrm{P}(w|c)$ is the probability of emission for the word w in a particular class c. The probabilities can be estimated by counting the relative frequencies of unigrams and bi-grams. To determine the optimal classes C for a number of classes M, we can adopt a maximum likelihood approach to find $\mathrm{C} = \arg\max_{\mathrm{C}} \mathrm{P}(W_1^N|\mathrm{C})$. It can be shown that the best possible grouping is that resulting from maximizing the average mutual information between adjacent clusters, given by:

$$\sum_{c,c'} \mathrm{P}(c,c') \times \log\left(\frac{\mathrm{P}(c,c')}{\mathrm{P}(c) \times \mathrm{P}(c')}\right)$$

The estimation of the language model is thus based on an agglomerative clustering procedure, which is used to build a tree hierarchy over the word class distributions. The algorithm starts with a set of leaf nodes, one for each of the word classes (i.e., initially one for each of the words), and then iteratively selects pairs of nodes to merge, greedily optimizing a clustering quality criteria based on the average mutual information between adjacent word clusters [Brown et al., 1992]. Each word is thus initially assigned to its own cluster, and the algorithm iteratively merges two classes so as to induce the minimum reduction on the average mutual information, stopping when the number of classes is reduced to the predefined number $|C|$.

For this work, to induce the generalized representations of words, we used an open-source[7] implementation of Brown's algorithm, following the description given by Turian et al. [2010]. This software was used together with a large collection of texts in Portuguese. These Portuguese texts correspond to a set of phrases that combines the CINTIL corpus of modern Portuguese [Barreto et al., 2006], with news articles published in the *Público*[8] newspaper, over a period of 10 years. We induced one thousand word groups, where each group has a unique identifier.

3.2 Feature Weighting

In this work, we also experimented with different ways to compute the feature weights to be used within our models, for both the textual terms and the word clusters, including binary values, term frequency scores, TF-IDF, and also with more sophisticated term weighting schemes, such as the Delta-TF-IDF and Delta-BM25 approaches previously discussed by Martineau and Finin [2009] and by Paltoglou and Thelwall [2010].

In the case of binary weights, each $w_{i,j}$ value is either zero or one, depending on whether the element i is present or not in document j.

In addition to binary values, another common approach is to use the frequency of occurrence of each element i within document j, most often also considering a logarithmic penalty.

TF-IDF is perhaps the most popular term weighting scheme, combining the individual frequency for each element i in a document j (i.e., the Term Frequency component, or TF), with the inverse frequency of element i in the entire collection of documents (i.e., the Inverse Document Frequency component, or IDF). When $n_i > 0$, the TF–IDF weight of an element i for a document j is given by the following formula:

$$\text{TF–IDF}_{i,j} = \log_2(1 + \text{TF}_{i,j}) \times \log_2\left(\frac{N}{n_i}\right)$$

In the formula, N is the total number of documents in the collection, and n_i is the number of documents containing the element i. The TF–IDF weight is 0 if $n_i \leq 0$.

The Delta–TF–IDF and Delta–BM25 schemes measure the relative importance of a term in two distinct classes. In the context of our regression problems, we have no binary classifications associated to each of the instances, but instead real values. We nonetheless considered two classes in order to determine the feature weights according to these schemes, by splitting the examples into those that have a target value greater or equal to the median target value in the training data, and those that are less or equal than the median.

[7] `github.com/percyliang/brown-cluster`

[8] `www.publico.pt`

The Delta–TF–IDF scheme extends the TF–IDF formulation, and localizes the estimation of IDF scores to the documents associated with each of the two classes, later subtracting the two values. Thus, the weight of an element i in a document j can be given as shown in the equation bellow, for the cases in which the $\mathrm{TF}_{i,j} > 0$:

$$\begin{aligned}\Delta\text{TF–IDF}_{i,j} &= \log_2(1 + \mathrm{TF}_{i,j}) \\ &\times \log_2\left(\frac{N_{pos} \times n_{i,neg}}{n_{i,pos} \times N_{neg}} + 1\right)\end{aligned}$$

In the formula, each N_c corresponds to the number of training documents respectively in collection c, and $n_{i,c}$ is the number of documents from collection c in which the term i occurs. In the context of this work, c can be *positive* for examples with a target value above the median, and *negative* for these bellow the median. According to a large set of experiments related to binary opinion classification, the approach named Delta–TF–IDF performs significantly better than the traditional TF–based or the binary weighting schemes [Martineau and Finin, 2009].

Paltoglou and Thelwall [2010] concluded that by additionally introducing smoothed localized variants of the original IDF functions, together with scaled or binary TF weighting schemes, the accuracy can be further increased. In the Delta–BM25 term weighting scheme, the weight of an element i for a document j is given by the following equation, where s is a smoothing constant that is usually set to 0.5:

$$\begin{aligned}\Delta\mathrm{BM25}_{i,j} =& \log_2(1+\mathrm{TF}_{i,j}) \\ &\times \log_2\left(\frac{(N_{pos}-n_{i,pos}+s)\times n_{i,neg}+s}{(N_{neg}-n_{i,neg}+s)\times n_{i,pos}+s}+1\right)\end{aligned}$$

3.3 Regression Models

One can use several types of regression models in order to address text-driven forecasting tasks. In this work, we compared linear regression models, using different types of regularization, against models based on ensembles of trees.

3.3.1 Linear Regression Methods

Considering a dataset $\{y_i, x_{i1}, ..., x_{ik}\}_{i=1}^{n}$ with n instances, and assuming that the relationship between the dependent variable y_i and the k-vector of features x_i is linear, we have that a linear regression model takes the following form:

$$\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{k1} \\ 1 & x_{12} & \cdots & x_{k2} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x_{1n} & \cdots & x_{kn} \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \\ \cdots \\ b_k \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \cdots \\ e_n \end{pmatrix}$$

In the formula, x_{ij} corresponds to the i-th feature of the j-th instance, the b_i parameters correspond to the regression coefficients, and e_j is an error that captures the difference between the actual observed responses y_i, and the prediction outcomes of the regression model. The formula can be rewritten in a matrix notation form:

$$y = Xb + e$$

There are several procedures that have been developed for parameter estimation and inference in linear regression. Linear Least Squares Regression (LSR) is the simplest and most widely used method for estimating the unknown parameters in a linear regression model. LSR minimizes the sum of the squared residuals $\mathrm{S}(b)$ between the data and the model, i.e., it minimizes $\sum_{i=1}^{n} e_i^2$. The squared residuals can be rewritten in matrix notation as $e'e$, where the apostrophe means that the matrix was transposed. Replacing e by $y - Xb$, we have that:

$$\begin{aligned}\mathrm{S}(b) &= \sum_{i=1}^{n} e_i^2 \\ &= (y - Xb)'(y - Xb) \\ &= y'y - y'Xb - b'X'y + b'X'Xb\end{aligned}$$

The condition for $\mathrm{S}(b)$ to be at a minimum is for the derivatives $\frac{\partial \mathrm{S}(b)}{\partial b} = 0$. The first term of the above equation does not depend on b, while the second and the third terms are equal in their derivatives, and the last term is a quadratic form of the elements b. Thus, we have that:

$$\frac{\partial \mathrm{S}(b)}{\partial b} = -2X'y + 2X'Xb$$

Equaling the differential equation to zero we get:

$$\begin{aligned}-2X'y + 2X'Xb &= 0 \\ X'Xb &= X'y \\ b &= (X'X)^{-1}X'y \\ b &= \left(\frac{1}{n}\sum_{i=1}^{n} x_i x_i'\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} x_i y_i\right) \\ b &= \arg\min_b ||y - Xb||^2\end{aligned}$$

In linear least squares regression, it is well know that when the number of training instances n is less than the number of features k, or if there are many correlated features, the regression coefficients can exhibit a high variance, and they tend to be unstable. In this case, regularized or penalized methods are needed to fit the regression model to the data, and to keep the variance of the regression coefficients under control.

The Ridge regression approach penalizes the size of the regression coefficients, by adding a l_2-penalty $||b||_2^2$ to the model. Thus, the Ridge regression coefficients are estimated as follows:

$$b = \arg\min_b ||y - Xb||^2 + \lambda||b||_2^2$$

In the formula $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. When $\lambda = 0$ we get the regular linear regression estimate, and when $\lambda = \infty$ we get $b = 0$.

Tibshirani [1996] proposed another penalized method, called Least Absolute Shrinkage and Selection Operator (Lasso), that shrinks some coefficients and sets others to zero. The Lasso method uses a l_1-penalty $||b||_1$ for model regularization, and tries to estimate the parameters b by minimizing the following formula:

$$b = \arg\min_b ||y - Xb||^2 + \lambda||b||_1$$

One of the main differences between Lasso and Ridge regression is that in Ridge regression, as the penalty is increased, all the parameters are reduced while still remaining non-zero, whereas with Lasso regularization, increasing the penalty will cause more of the parameters to be driven to zero. Hence, Lasso models tend to be sparse, in the sense that they use less features. However, one limitation of the Lasso is that if $k > n$, the Lasso selects at most n variables, i.e., the number of selected variables is bounded by the number of training examples. Another limitation of the Lasso method occurs when there is a group of highly correlated variables. In this case, the Lasso tends to select one variable from a group, ignoring the others. To solve these problems Zou and Hastie [2005] proposed the Elastic Net approach, that combines the l_1 and l_2 regularizations with weights λ_1 and λ_2, respectively. The estimates from the Elastic Net method are defined by the following formula:

$$b = \arg\min_b ||y - Xb||^2 + \lambda_1||b||_1 + \lambda_2||b||_2^2$$

The Elastic Net method tends to result in better predictions when the variables are considerably correlated. The method removes the limitation on the number of selected variables, encourages grouping effects, and stabilizes the l_1 regularization approach.

Various methods have been proposed for estimating linear regression models with Ridge, Lasso, and Elastic Net regularization, including cyclical coordinate descent [Kim and Kim, 2006], or other convex optimization methods based on iterative computations [Boyd and Vandenberghe, 2004], such as SpaRSA [Wright et al., 2009]. In this study, we used the implementation available from the scikit-learn Python package.

3.3.2 Ensemble Learning Methods

Ensemble learning methods attempt to combine multiple models, usually relatively simple models based on trees, to achieve better predictive performance [Sewell, 2011]. In our experiments, we used two types of ensemble regression methods, namely Random Forests, and Gradient Boosted Regression Trees, again as implemented within the scikit-learn Python package.

The Random Forest is an ensemble-learning method based on decision trees, for classification and regression [Breiman, 2001]. This method combines the idea of bagging, developed by Breiman [1996], with a random selection of features. In brief, we have that the Random Forest method builds a collection of de-correlated trees, and then averages them. This approach has been argued to have a similar performance to boosting, but it is simpler to train and tune. The main objective in Random Forests is to reduce variance, by reducing the correlation between variables. This is achieved by the random selection of variables. Let N be the number of training cases, and let M be the number of instances used for model training. The algorithm proceeds is as follows:

1. Choose a training set for each tree in the ensemble, by sampling n times with replacement from all N available training instances. Use the remaining instances to estimate the error of the tree, when making predictions.

2. For each node of the tree, select m variables at random to support the decision at that node, and calculate the best split for the tree, based on these m variables and using the training set from the previous step.

3. Each tree grows to the largest extend, and no pruning is performed. The CART algorithm proposed by Breiman et al. [1984] is used for growing the trees in the ensembles.

The above steps are iterated to generate a set of trees. When making a prediction decision, the average vote of all trees is reported as the prediction. Each tree votes with a weight corresponding to its performance on the subset of the data that was left out during training.

As for Gradient Boosted Regression Trees (GBRT), this is another ensemble method instead based on the idea of boosting, supporting loss functions for regression such as the sum of the squared errors [Friedman, 2001]. A GBRT model consists of an ensemble of weak prediction models, typically decision tress (i.e., CART trees, similar to those that are used in the case of Random Forest regression) of the following form, where $\mathrm{h}_m(X)$ are the basis functions, which are usually called weak learners in the context of ensemble methods based on boosting.

$$y = \mathrm{F}(X) = \sum_{m=1}^{M} \mathrm{h}_m(X)$$

Similarly to most boosting algorithms, GBRT builds the additive model in a forward stage-wise procedure. In accordance with the empirical risk minimization principle, the method tries to minimizes the average value of a loss function over the training set. It does so by starting with a model, consisting of a constant function F_0, and incrementally expanding it in a greedy fashion, according to the following equation:

$$\mathrm{F}_m = \mathrm{F}_{m-1}(X) - \gamma_m \mathrm{h}_m(X)$$

At each stage, a decision tree $\mathrm{h}_m(X)$ is chosen to minimize the considered loss function (i.e., the sum of the squared errors) given the current model F_{m-1} and its fit $\mathrm{F}_{m-1}(X)$ (i.e., the trees h_m learned at each stage are grown by using pseudo-residuals as the training set). The initialization for model F_0 is commonly chosen with basis on the mean of the target values, and the multiplier γ_m is found at each stage by solving the following optimization problem, where L is the loss function:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} \mathrm{L}(y_i, \mathrm{F}_{m-1}(x_i) + \gamma \mathrm{h}_m(x_i))$$

The optimization problem is typically addressed numerically via a steepest descent procedure [Boyd and Vandenberghe, 2004].

4 Experimental Validation

This work explored text-driven forecasting in three distinct domains with different characteristics, namely by considering information about hotels, restaurants and their prices in Portugal, or about movie reviews together with the corresponding box-office revenues. For hotels and restaurants, we crawled textual descriptions from Lifecooler[9]. For each restaurant, the information available from this site includes the name, a textual description, the menu, the specialities, the type of restaurant, the average meal price, and the location, which includes the city name and the district. For the hotels, the information available includes the name, a textual description, the location, and the price over the low and high tourist seasons. For the case of movies, we used review data from Portal do Cinema[10], and metadata from Instituto do Cinema e do Audiovisual[11], for movies that were released from 2008 to 2013. The available information includes the movie name, the distributor, the producer, the number of screens on which the movie played during the first week, the gross revenue during the first week, the number of viewers, and the release date. Each review includes the movie name, the textual review, and a star rating on a scale from 0 to 5. Only movies found on both websites were included in our dataset, and we ended up with a set of 502 movies. The main characteristics of all three datasets are presented in Table 1. For hotels and restaurants, statistics over the target values are shown in Euros, while for the case of movies they are shown in thousands of Euros.

The datasets differ in several aspects, such as in the number of documents, and the in distribution for the values to be predicted. The distribution of the target values for all three domains is shown through violin plots on Figure 1.

All the available text was used in our experiments (e.g., for hotels, we used the hotel name and textual description, while for restaurants we used the restaurant name, the textual description, the specialities, and the menus). For movies, we used the movie name, and the textual review). In addition to textual features, in some of the experiments we used location features, for hotels and restaurants, type features for restaurants, or the number of screens that the movie was played. The location features (i.e., the administrative districts) can naturally influence how expensive is a hotel or a restaurant. For instance, as we can see in Figure 2, the districts with the most expensive hotels and restaurants are Lisbon and Faro, and those same districts are the ones showing the highest variation in the corresponding prices.

[9] www.lifecooler.com

[10] www.portal-cinema.com

[11] www.ica-ip.pt

Table 1: Statistical characterization for the three different datasets.

	Hotels High	**Hotels Low**	**Restaurants**	**Movies**
Number of textual descriptions	2656	2656	4677	502
Term vocabulary size	9932	9932	19421	28720
Average number of terms per text	35	35	47	346
Minimum target value	10.00	10.00	4.50	0.93
Maximum target value	3000.00	1200.00	100.00	1437.71
Mean target value	95.92	71.48	18.10	162.75
Median target value	75.00	60.00	15.00	73.56
Standard deviation on target values	93.42	51.67	8.41	229.21

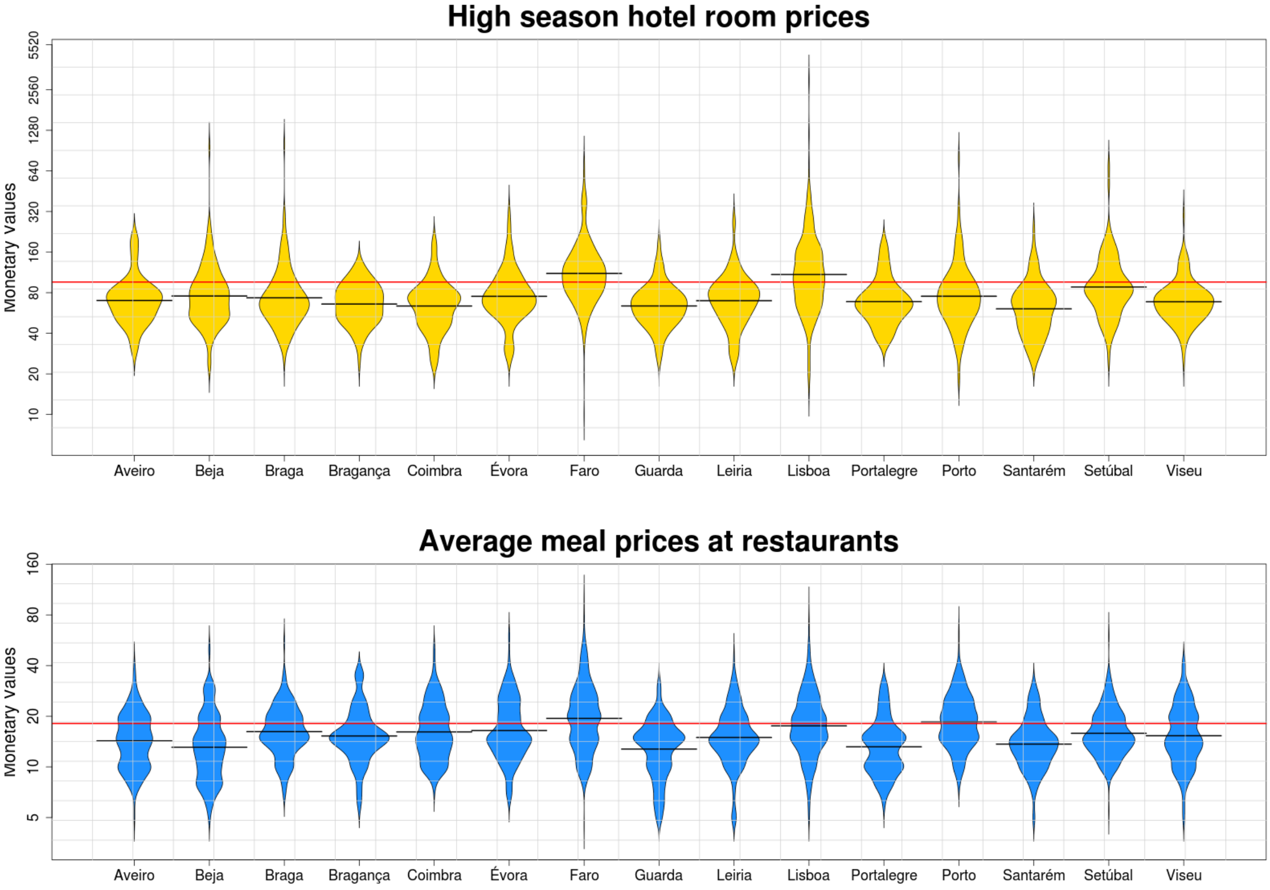

Figure 2: Distribution for the monetary values to be predicted, per district in Continental Portugal.

All our experiments were done with a 10-fold cross validation methodology, and the quality of the obtained results was measured using evaluation metrics such as the Mean Absolute Error, and the Root Mean Squared Error.

The Mean Absolute Error (MAE) is a measure that compares forecasts against their eventual outcomes, essentially corresponding to an average of the absolute errors, as shown in Equation 1. The Root Mean Squared Error (RMSE) is another measure of the accuracy of a predictor, computed as the square root of the mean of the squares of the errors, as shown in Equation 2.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \qquad (1)$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2} \qquad (2)$$

Considering a dataset $\{y_i, x_{i1}, ..., x_{ik}\}_{i=1}^{n}$, where x_{ik} corresponds to the inputs, where y_i corresponds to the true outputs, and having $\hat{y_i}$ corresponding to the predicted outputs, one can easily see that the previous metrics estimate errors in the same units of measurement as the target value, i.e., in Euros or in thousand of Euros, in the case of the experiments reported here.

Table 2: Results for the first experiment, with a representation based on TF-IDF.

	Hotels High		Hotels Low		Restaurants		Movies	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Average	44.06	93.65	28.58	51.86	6.09	8.41	161.15	247.87
Median	39.96	95.69	26.59	52.92	5.80	8.97	140.48	264.29
Ridge Regression	45.87	72.94	30.38	42.41	6.40	6.83	127.70	201.52
Lasso	35.78	72.96	24.27	43.60	4.59	6.57	183.01	268.33
Elastic Net	**34.63**	**70.86**	**23.25**	**41.97**	**4.27**	**6.20**	**127.55**	**192.70**
Random Forest	**34.25**	**74.13**	**23.17**	**44.25**	**4.40**	**6.56**	**135.89**	**211.77**
Gradient Boosting	37.91	79.94	25.18	47.09	4.65	7.02	166.74	269.95

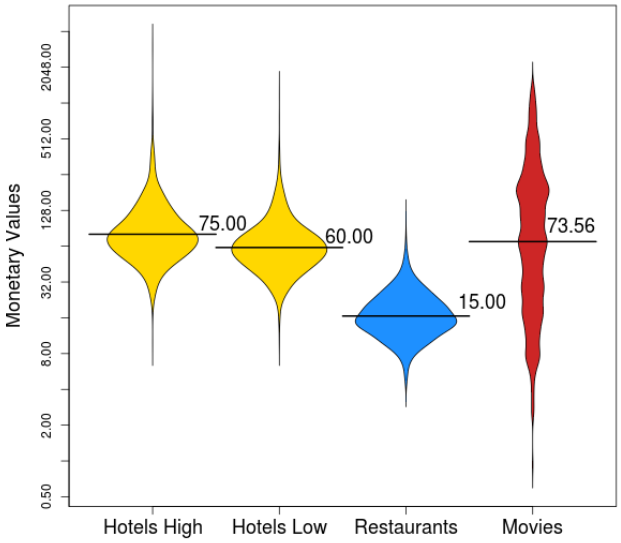

Figure 1: Distribution of monetary values, in the hotels, restaurants, and movies datasets.

4.1 Experimental Results

In a first experiment, we tried to predict room prices for hotels, the average meal prices for restaurants, or the movie box-office revenues, by using only the textual contents. In this task, we compared our regression models against baselines such as the average and median value, considering representations based on the most popular term weighting scheme, i.e., TF-IDF. As we can see in Table 2, regression models using the text achieved better results than the considered baselines. We also found that, of all the models that were used, the best results were achieved with the Elastic Net method. The best ensemble model is given by the Random Forest approach.

In a separate set of experiments, we attempted to analyse the importance of the different features corresponding to textual tokens, seeing their relative differences in terms of the contribution to predicting the target values. This was made for the case of models based on Random Forests, or based on linear regression with Elastic Net regularization, using feature weights computed with the TF-IDF approach.

In the case of linear regression models with Elastic Net regularization, we inspected the feature weights (i.e., the regression coefficients) of our learned models, averaging the weights of each feature over the multiple folds of our cross-validation experiments. In the case of Random Forest regression models, the relative rank (i.e., the depth) of a feature that is used as a decision node in a tree can be used to assess the relative importance of that feature, with respect to the predictability of the target variable. Features that are used at the top of the trees contribute to the final prediction decision of a larger fraction of the input samples and, thus, the expected fraction of the samples they contribute to can be used as an estimate of the relative importance of the features. By averaging those expected activity rates over the several trees of the Random Forest ensemble, and by averaging also over the multiple folds of our cross-validation experiments, one can estimate a feature importance score.

Figure 3 plots the 20 most important features in terms of either the linear regression coefficients (i.e., the 10 features with the highest positive or negative values) or in terms of the relative rank of the decision nodes, for the case of models predicting hotel room prices in the high and low seasons. As expected, terms such as *sheraton, luxo* or *resort* seem to indicate higher prices, whereas terms such as *pensão* or *hostel* are associated with lower prices. Figure 4 also plots the 20 most important features, but in this case for models predicting meal prices at restaurants (i.e., the plots on the left), and for models predicting movie box-office results. In the case of restaurants, terms such as *hotel* or *michelin* seem to be highly discriminative, whereas in the case of movie box-office results, terms such as *saga* or *terror* seem to provide the best clues.

Tables 3 and 4 show a comparison of the results obtained with the best models, i.e., linear regression with Elastic Net regularization

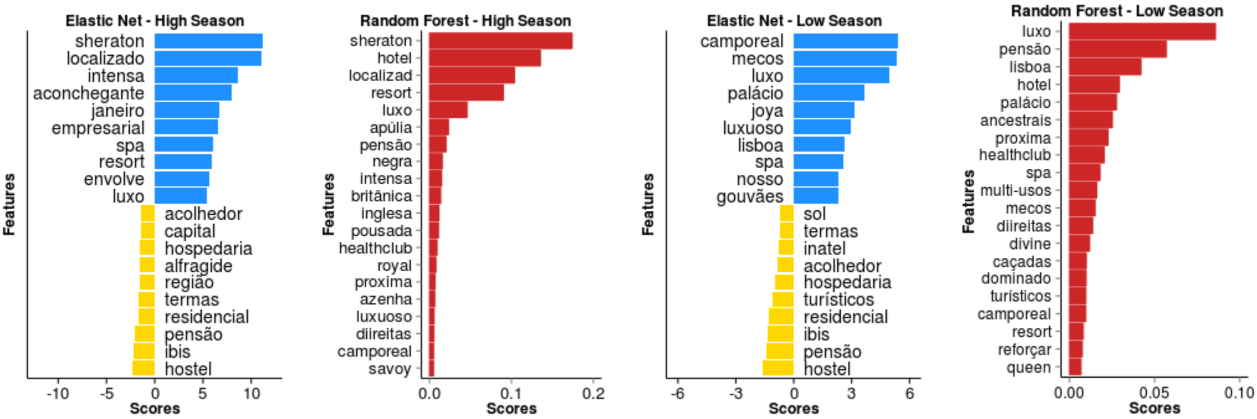

Figure 3: The 20 most important features for the case of predicting hotel room prices.

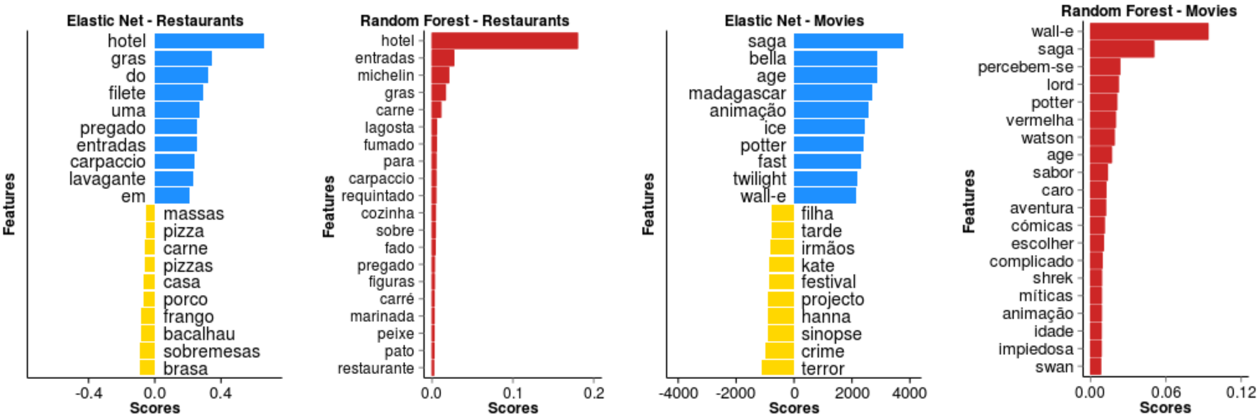

Figure 4: The 20 most important features for the case of predicting meal prices at restaurants, or in the case of predicting movie box-office results.

Table 3: Results with Elastic Net models using different feature weighting schemes.

	Hotels High		**Hotels Low**		**Restaurants**		**Movies**	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Binary	40.91	77.49	27.14	46.64	5.94	8.22	133.01	199.34
Term Frequency	51.18	86.10	30.34	48.64	5.42	6.31	209.50	279.51
TF-IDF	34.63	70.86	23.25	41.97	4.27	6.20	127.55	192.70
Delta-TF-IDF	**34.55**	**70.63**	24.33	41.77	4.36	6.62	131.59	194.37
Delta-BM25	34.70	72.82	**23.21**	**40.24**	**4.22**	**6.14**	**127.41**	**191.08**

and Random Forest regression, respectively, using each of the representations for the textual contents that were described in Section 3.3. The richer document representation is perhaps Delta-BM25, although Delta-TF-IDF achieved very similar or even better results on the case of models based on the Random Forest approach.

In addition to the textual contents, we considered other metadata elements, such as the location for hotels and restaurants, the type of restaurant, or the number of screens on which the movie was shown in the opening weekend. We also experimented with adding word clusters to the representation of the textual contents.

Metadata properties such as locations or the type of restaurant are represented as binary values (e.g., one for each administrative district), where a single position takes the value of 1 depending on the location or the type corresponding to the textual description. Thus, the final vector used in the experiment is the concatenation of the feature vector derived from the text (i.e., using just the individual terms, or the terms plus the corresponding word clusters), and the vector associated to the metadata. For the case of movies, we added the number of screens as one of the dimensions on the feature vector.

Table 5 lists the corresponding results when

Table 4: Results with Random Forest models using different feature weighting schemes.

	Hotels High		Hotels Low		Restaurants		Movies	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Binary	36.56	79.06	24.28	46.19	4.83	7.08	135.68	214.60
Term Frequency	36.62	77.09	25.53	46.69	5.37	6.51	137.92	207.82
TF-IDF	34.25	74.13	**23.17**	44.25	4.40	6.56	135.89	211.77
Delta-TF-IDF	**34.12**	**73.43**	24.91	44.04	**4.32**	6.85	**130.59**	**209.49**
Delta-BM25	34.47	73.55	23.19	**43.45**	4.63	**6.52**	134.84	210.24

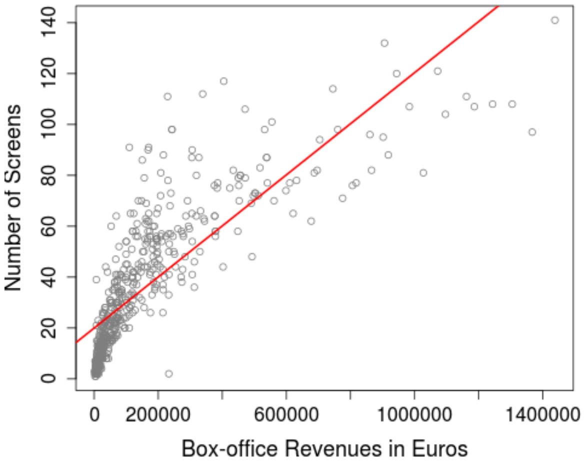

Figure 5: Box-office revenues versus the number of screens on which the movie was shown.

predicting hotel room prices with different feature sets. The combination of the textual contents, the metadata, and the word clusters achieved the best performance, both in term of MAE and RMSE, by using the Elastic Net approach. When using Random Forests, the combination of text and location yielded better results.

Table 6 shows the results for predicting average meal prices for restaurants, using different feature sets. With the Elastic Net method, the experiment that considered only the word features produced better results, and with the Random Forest approach, the combination of text and location produced better results.

Finally, Table 7 presents the corresponding results when predicting movie box-office revenues. The number of screens has a strong influence on the results. With both types of regression models, the experiment that yielded better results is clearly the one involving the combination of text and the number of screens. The number of screens and the box-office revenues are indeed highly correlated, as shown in Figure 5.

For restaurants and hotels, we also compared our results against baselines such as predicting the training set average value, and the average value per location. In the case of movies, we attempted to use just the number of screens. These results are reported in Table 8. We achieved better results by predicting average value per location, when compared to using the average over the entire training datasets.

In sum, and as reported in Table 8, we can conclude that regression models using textual contents indeed result in accuracy gains for the considered forecasting tasks.

5 Conclusions and Future Work

This paper presented an experimental study on the subject of making predictions with textual contents written in Portuguese, using documents from three distinct domains.

The specific tasks addressed in the paper involved (i) predicting room prices for hotels in Portugal, both in the high and low touristic seasons, using textual descriptions collected from a well known Web portal, (ii) predicting average meal prices in restaurants located in Portugal, using textual descriptions for the restaurants and their menus, as collected from the same Web portal, and (iii) predicting movie box-office results in the first week of exhibition, as reported by Instituto do Cinema do Audiovisual, for movies exhibited in Portugal and using textual reviews from another well-known Web portal. We specifically report on experiments using different types of regression models, using state-of-the-art feature weighting schemes, and using features derived from cluster-based word representations. Our experiments clearly show that prediction models using the textual information achieve better results than baselines such as the average value for the target variable, and that using richer document representations (i.e., using Brown clusters and the Delta-TF-IDF feature weighting scheme) may result in slight performance improvements.

Despite the interesting results, there are also many ideas for future work. Given that we only had access to relatively small training datasets, we believe that one interesting path for future work concerns evaluating semi-supervised learn-

Table 5: Results for predicting hotel room prices with different feature sets.

		Only Text		+WClusters		+Location		All	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Low Season	**Elastic Net**	**23.21**	**40.24**	23.32	42.55	23.57	43.05	23.40	42.83
	Random Forest	23.19	43.45	23.52	45.03	**23.18**	**43.06**	23.66	44.00
High Season	**Elastic Net**	34.70	72.82	**34.33**	**70.12**	34.75	70.46	34.53	70.81
	Random Forest	34.47	73.55	35.22	74.20	34.38	76.46	**34.07**	**74.63**

Table 6: Results for predicting restaurant prices with different feature sets.

	Only Text		+WClusters		+Type		+Location		All	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Elastic Net	**4.22**	**6.14**	4.90	7.04	4.88	7.03	4.87	7.02	4.94	7.04
Random Forest	4.63	6.52	4.45	6.66	4.36	6.59	**4.33**	**6.52**	4.41	6.65

Table 7: Results for predicting movie box-office revenues with different feature sets.

	Only Text		+WClusters		+Screens		All	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Elastic Net	127.91	191.08	109.87	179.09	79.06	125.45	**70.37**	**126.03**
Random Forest	130.59	209.49	136.24	208.22	**66.02**	**137.78**	79.65	128.53

Table 8: Overall results for the different forecasting tasks.

	Hotels High		Hotels Low		Restaurants		Movies	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Average	44.06	93.65	28.58	51.86	6.09	8.41	161.15	247.87
Average per Location	40.57	90.80	28.11	50.91	5.81	8.11	–	–
Number of Screens	–	–	–	–	–	–	83.45	131.89
Best Regression Model	**34.07**	**70.12**	**23.18**	**40.24**	**4.22**	**6.14**	**66.02**	**122.29**

ing techniques, capable of levering large amounts of non-labeled data for text-driven forecasting. It also seems reasonable to assume that the cues to correctly estimating a given target value, based on a textual document, may lie in a handful of the document's sentences. Yogatama and Smith [2014] have, for instance, introduced a learning algorithm that exploits this intuition trough a carefully designed regularization approach (i.e., a sparse overlapping group Lasso, with one group for every bundle of features occurring together in a training-data sentence), showing that the resulting method can significantly outperform other approaches (e.g., standard Ridge, Lasso, and Elastic Net regularizers) on many different real-world text categorization problems. Finally, given the relative success of document representations using Brown clusters, we would also like to experiment with other types of representations based on distributional similarity, such as the word embeddings proposed in the study by Mikolov et al. [2013].

Acknowledgements

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT), through the project grant with reference UTA-Est/MAI/0006/2009 (REACTION), as well as through the INESC-ID associate laboratory multi-annual funding from the PIDDAC programme (PEst-OE/EEI/LA0021/2013).

We would like to express our sincere gratitude to the colleagues from the REACTION project, for their assistance and insightful comments.

References

J. Scott Armstrong and Fred Collopy. Regression Methods for Poisson Process Data. *Journal of the American Statistical Association*, 82(399), 1987.

Florbela Barreto, António Branco, Eduardo Ferreira, Amália Mendes, Maria Fern, A Bacelar Do Nascimento, Filipe Nunes, and João Ricardo Silva. Open resources and tools for the shallow processing of portuguese: The

TagShare project. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2006.

Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 1(2), 2011.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Leo Breiman. Bagging Predictors. *Machine Learning*, 24(2), 1996.

Leo Breiman. Random Forests. *Machine Learning*, 45(1), 2001.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.

Peter F. Brown, Peter V. de Souza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 1992.

Victor Chahuneau, Kevin Gimpel, Bryan R. Routledge, Lily Scherlis, and Noah A. Smith. Word salad: Relating food prices and descriptions. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, 2012.

Mats Dahllöf. Automatic prediction of gender, political affiliation, and age in swedish politicians from the wording of their speeches—a comparative study of classifiability. *Literary and Linguistic Computing*, 27(2), 2012.

Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Proceeding of the Conference on Neural Information Processing Systems*, 1997.

Jerome Fridman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2008.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 2001.

Yancheng Hong and Steven Skiena. The wisdom of bookies? sentiment analysis vs. the NFL point spread. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*, 2010.

Mahesh Joshi, Dipanjan Das, Kevin Gimpel, and Noah A. Smith. Movie reviews and revenues: An experiment in text regression. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

Yongdai Kim and Jinseog Kim. Gradient Lasso for feature selection. In *Proceedings of the International Conference on Machine Learning*, 2006.

Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. Predicting risk from financial reports with regression. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

Kevin Lerman, Ari Gilder, Mark Dredze, and Fernando Pereira. Reading the markets: forecasting public opinion of political candidates by news analysis. In *Proceedings of the International Conference on Computational Linguistics*, 2008.

Xueming Luo, Jie Zhang, and Wenjing Duan. Social media and firm equity value. *Information Systems Research*, 24(1), 2013.

Justin Martineau and Tim Finin. Delta TF-IDF: An improved feature space for sentiment analysis. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*, 2009.

Peter McCullagh. Regression models for ordinal data. *Journal of Royal Statistical society*, 42 (2), 1980.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems*, 2013.

Lewis Mitchell, Morgan R. Frank, Kameron Decker Harris, Peter Sheridan Dodds, and Christopher M. Danforth. The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place. *PLOS ONE*, 8(5), 2013.

Brendan O'Connory, Ramnath Balasubramanyany, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the AAAI International*

Conference on Weblogs and Social Media, 2010.

Georgios Paltoglou and Mike Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2010.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2), 2008.

Kira Radinsky. Learning to predict the future using web knowledge and dynamics. *ACM SIGIR Forum*, 46(2), 2012.

Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems*, 27(12), 2009.

H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Megha Agrawal Gregory J. Park, Shrinidhi K. Lakshmikanth, Sneha Jha, Martin E. P. Seligman, Lyle Ungar, and Richard E. Lucas. Characterizing geographic variation in well-being using tweets. In *Proceeding of the AAAI International Conference on Weblogs and Social Media*, 2013.

Martin Sewell. Ensemble methods. Technical Report RN/11/02, University College London Departament of Computer Science, 2011.

Noah A. Smith. Text-Driven Forecasting, 2010.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B*, 58(1), 1996.

Seshadri Tirunillai and Gerard J. Tellis. Does chatter really matter? Dynamics of User-Generated Content and Stock Performance. *Information Systems Research*, 31(2), 2012.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representation: a simple and general method for semi-supervised learning. In *Proceedings of the Annual Meeting of the Association for Computional Linguistics*, 2010.

Stephen J. Wright, Robert D. Nowak, and Mário A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7), 2009.

Tae Yano and Noah A. Smith. What's worthy of comment? Content and Comment Volume in Political Blogs. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*, 2010.

Tae Yano, William W. Cohen, and Noah A. Smith. Predicting response to political blog posts with topic models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

Tae Yano, Noah A. Smith, and John D. Wilkerson. Textual predictors of bill survival in congressional committees. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2012.

Dani Yogatama and Noah A. Smith. Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proceedings of the International Conference on Machine Learning*, 2014.

Hui Zou and Trevor Hastie. Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society Series B*, 67(5), 2005.