TÉCNICO LISBOA

Making Predictions with Textual Contents

Indira Gandi Mascarenhas de Brito

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Bruno Emanuel da Graça Martins

Examination Committee

Chairperson: Prof. Mário Jorge Costa Gaspar da Silva
Supervisor: Prof. Bruno Emanuel da Graça Martins
Member of the Committee: Prof. David Manuel Martins de Matos

April 2014

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 document presents an experimental study on the subject of making predictions with textual contents written in Portuguese, using documents from three distinct domains. I 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.

Through controlled experiments, I have shown that prediction models using the textual information achieve better results than simple baselines such as taking the average value over the training data, and that richer document representations (i.e., using Brown clusters and the Delta-TF-IDF feature weighting scheme) result in slight performance improvements.

Keywords: Text-Driven Forecasting, Supervised Learning of Regression Models, Word Clustering, Feature Engineering for NLP Applications

Sumário

A previsão de quantidades do mundo real com base em informação proviniente de descrições textuais atraiu recentemente um interesse significativo enquanto problema de investigação, embora os estudos anteriores na área se tenham concentrado em aplicações que envolvem apenas textos no idioma Inglês.

Este documento apresenta um estudo experimental sobre a realização de previsões com base em conteúdos textuais escritos em Português, envolvendo o uso de documentos associados a três domínios distintos. Eu relato especificamente experiências utilizando diferentes tipos de modelos de regressão, usando esquemas de pesagem para as características do actual estado da arte, e usando características derivadas de representações para as palavras baseadas no agrupamento automático das mesmas.

Através de experiências controladas, desmonstrei que modelos preditores usando informação textual atingem melhores resultados, quando comparados com abordagens simples tais como realizar as previsões com base no valor médio dos dados de treino. Demonstrei ainda que as representações de documentos mais ricas (ou seja, usando o algoritmo de Brown para o agrupamento automático de palavras, e o esquema de pesagem das características denomindo Delta-TF-IDF) resultam em ligeiras melhorias no desempenho.

Palavras Chave: Previsões com Base em Texto, Modelos de Regressão, Agrupamento Automático de Palavras Semelhantes, Engenharia de Caracteristicas para Aplicações em PLN

Acknowledgements

This work was 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 multi-annual funding from the PIDDAC programme (PEst-OE/EEI/LA0021/2013).

This work would not have been possible without the enormous contribution of all friends and colleagues that were present during the development of this project. I would like to express my thanks to these persons, for the friendship, encouragement, understanding, and wisdom that I had the privilege of receiving.

Firstly, I want to thank God for giving me life and health. Many thanks also to my supervisor, Dr. Bruno Martins, who read through my numerous revisions and helped in the course of the project.

To my parents Francisco Brito and Maria da Conceição Mascarenhas, for the emotional and financial support. Thanks for always believing in me.

To my brothers Diva Brito and Djeivy Brito, for their love and support. To my boyfriend, Gelton Delgado, my partner of all time. To my uncles, Manuel and Fatima, who have helped me since I arrived in Portugal. Finally, to all my Portuguese and Cape Verdean friends, specifically Aurea, Any, Elio, Elizangela, Lenise, Tatiana, Lilian, Rodney, Dirce, and Helton.

I would also like to express my sincere gratitude to the colleagues from the aforementioned REACTION project, for their assistance and insightful comments.

Thanks to all.

Contents

List of Tables

List of Figures

Chapter 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 (Radinsky, 2012; Smith, 2010). Well-known examples of previous studies include using textual contents for making predictions about stock or market behavior (Bollen *et al.*, 2011; Lerman *et al.*, 2008; Luo *et al.*, 2013; Schumaker & Chen, 2009; Tirunillai & Tellis, 2012), sports betting market results (Hong & Skiena, 2010), product and service sales patterns (Chahuneau *et al.*, 2012; Joshi *et al.*, 2010), government elections, legislative activities and general political leans (Dahllöf, 2012; Yano *et al.*, 2012), or general public opinion polls (Mitchell *et al.*, 2013; O'Connory *et al.*, 2010; Schwartz *et al.*, 2013). However, most previous work in the area has focused on applications over the English language.

My MSc thesis addressed the task 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. My research 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, my study also introduces some technical novelties in relation to most previous work in the area, namely by experimenting with (i) state-of-the-art Information Retrieval

[1] http://scikit-learn.org

feature weighting schemes such as Delta-TF-IDF or Delta-BM25, and (ii) features derived from cluster-based word representations such as those provided by Brown's clustering algorithm.

This chapter presents the main objectives of my MSc thesis, describing the research hypothesis and the evaluation methodology that I used in my work, as well as the main contributions that were achieved. The chapter ends with a description for the organization of this dissertation.

1.1 Thesis Proposal and Methodology

My MSc research project attempted to validate the claim that **text-driven forecasting problems can also be effectively handled with basis on contents written in Portuguese.**

I specifically tried to answer several questions regarding the problem of text-driven forecasting, such as those referenced below:

- Is it possible to get a good predictive performance with texts written in Portuguese? Are the results in this case comparable against those achieved for the English language?

- How do models based on ensembles of trees compare against linear regression models with state-of-the-art regularization schemes, on these particular tasks?

- How do different textual features (e.g., words or cluster-based features derived from word co-occurrences and representing latent topics) affect the performance of the models?

- Can a better performance be achieved through state-of-the-art feature weighting approaches such as Delta-TF-IDF or Delta-BM25? And how can these feature weighting methods be adapted to regression problems?

In order to answer the aforementioned questions, I used a methodology based on controlled experiments with textual resources associated to real-world quantities. I implemented a software framework supporting the realization of experiments, and I specifically collected textual contents from three distinct domains, namely:

- Descriptions for hotels in Portugal from a well-known Web portal named Lifecooler, associated to average room prices in the high and low seasons for tourists;

- Descriptions for restaurants and the corresponding menus from the same Web portal in the previous item, associated to the average meal prices;

- Movie reviews from a Web site named Portal do Cinema, together with official numbers for the corresponding box-office results for the first week of exhibition, as available from Instituto do Cinema e do Audiovisual.

Using the previous three datasets, I performed experiments using different types of regression models, different feature weighting schemes, and different feature sets. In order to evaluate the quality of the results, so as to compare the different forecasting models, I used a 10-fold cross validation technique, together with common evaluation metrics such as the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE).

There are several software frameworks that can support the realization of experiments with machine learning algorithms. For this work, I used the implementations of state-of-the-art learning methods that are available in an open source Python machine learning library named scikit-learn. This package was thus integrated with the rest of the software that I developed.

1.2 Contributions

The main contributions resulting from this work are as follows:

- I created a software framework that includes programs for collecting data from the Web and for data processing, as well as methods for computing different feature weighting schemes, and for feature analysis through visual inspection. This software framework can be now exploited in other research studies focusing on text-driven forecasting, adapting it to the domain being considered.
- Previous text-driven forecasting studies have focused on applications involving only textual contents written in English. This thesis introduces the novelty of using textual contents written in Portuguese. I specifically used Portuguese datasets that were collected from well-know web sites, such as Lifecooler and Portal do Cinema. The obtained results show that it is possible to address different types of forecasting tasks with texts written in Portuguese. In all three domains, models using features derived from the text have significantly outperformed baselines such as predicting the average value.
- Previous work in the context of text-driven forecasting has mostly used linear regression models. In my work, I compared linear regression models with models based on ensembles of trees, such as Random Forests for regression and Gradient Boosted Regression Trees. I also experimented with linear regression models using different types of model regularization approaches. The best results were achieved with linear regression using the

Elastic Net regularization method. The Random Forest approach achieved slightly inferior results, and this was the best approach in the case of ensemble learning methods.

- I also experimented with features derived from cluster-based word representations. I specifically used an open-source implementation of Brown's word clustering algorithm. To induce the representations of words, I used Portuguese texts corresponding to a very large set of phrases that combines the CINTIL corpus of modern Portuguese, with news articles published in the *Público* newspaper. For instance in the case of hotel high season prices, when using the Elastic Net regularization approach, the combination of the textual terms and the word clusters achieved slightly better results, when compared with a model that used just the textual terms. However, in some of our experiments, using word clusters actually lead worse results.

- My research introduced the usage of state-of-the-art feature weighting schemes, such as Delta-BM25 or Delta-TF-IDF, comparing their performance against more traditional feature weighting functions. The richer representation is perhaps Delta-BM25, although the obtained results with Delta-TF-IDF scheme, or with TF-IDF alone, were very similar in terms of the prediction accuracy.

It is also worth mentioning that this dissertation was summarized in a paper that was submitted to a journal focusing on the automatic processing of Iberian languages, called Linguamática[1]. A second shorter paper, describing initial experiments with applications to the Portuguese language, was also accepted at the 3rd Spanish Conference on Information Retrieval (CERI 2014).

1.3 Document Organization

The rest of this dissertation is organized as follows:

- Chapter 2 presents previews work related with the task of text-driven forecasting, describing the datasets, the techniques used by the different authors, and the main results that were reported on the corresponding articles.

- Chapter 3 details the main contributions from the research made on the context of this thesis, presenting the regression techniques that were considered, as well as the approaches taken for representing the textual contents as feature vectors.

- Chapter 4 presents the experimental evaluation of the proposed methods. First, I present the characteristics of each of the three domain datasets, as well as the distribution of the

[1] `http://www.linguamatica.com/`

values to be predicted. Then, I present the results achieved in my experiments, comparing the results across different types of models, with different feature weighting schemes, and with different features for representing the instances.

- Chapter 5 summarizes the main conclusions of this work. It presents an overview on the contributions, and provides some guidelines for work that can be developed in the future, with basis on the results achieved through this thesis.

Chapter 2

Previous and Related Work

In this chapter, I present the most relevant previous work that addressed challenges related to text-driven forecasting. Particular emphasis is given to the previous work developed by Noah Smith and his colleagues from the Language Technology Institute at Carnegie Mellon University. For each of the considered previous studies, I describe the datasets, the techniques used by the authors, as well as the main results that were reported on the corresponding articles.

2.1 Work in Text-Driven Forecasting from Noah Smith et al.

Taking inspiration from recent research on sentiment analysis (Pang & 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 the textual contents 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 textual contents from political blog posts, to try predicting the response that they will gather from the readers (Yano & Smith, 2010).

4. The interpretation of a day's microblog feeds, in order to try predicting the public's opinion about a particular issue (O'Connory *et al.*, 2010).

5. The interpretation of food writings, as given on restaurant descriptions, on restaurant menus, and on customer reviews, to try predicting average meal prices and costumer rantings for the restaurants (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 of different approaches can be 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. Smith considered making predictions about the box-office with basis on the text from these reviews, right after they are produced by expert critics. He 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*[1], 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*[2] 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 reviews made before the release date of the movie.

[1] http://www.metacritic.com
[2] http://www.the-numbers.com

Smith described the application of linear regression modeling with state-of-the-art Elastic Net regularization (Fridman *et al.*, 2008; Zou & Hastie, 2005). The model was trained on 988 examples released from 2005-2007, and it was evaluated by forecasting the 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 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 & 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 & 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 prediction. 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 & 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[1] 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*[2]. From this website, 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 ratings in a 5-star scale), for most of the restaurants, were collected from a service named *Yelp*[3].

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

[1] http://www.gallup.com/poll/122840/gallup-daily-economic-indexes.aspx
[2] http://www.allmenus.com
[3] http://www.yelp.com

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 exceed the value of 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., if it was above or below the average rating. The obtained Accuracy was of 87%.

The authors also performed a fine-grained analysis of the sentiments expressed in the review texts. The aim was to see the contribution of this particular aspect to the price range prediction

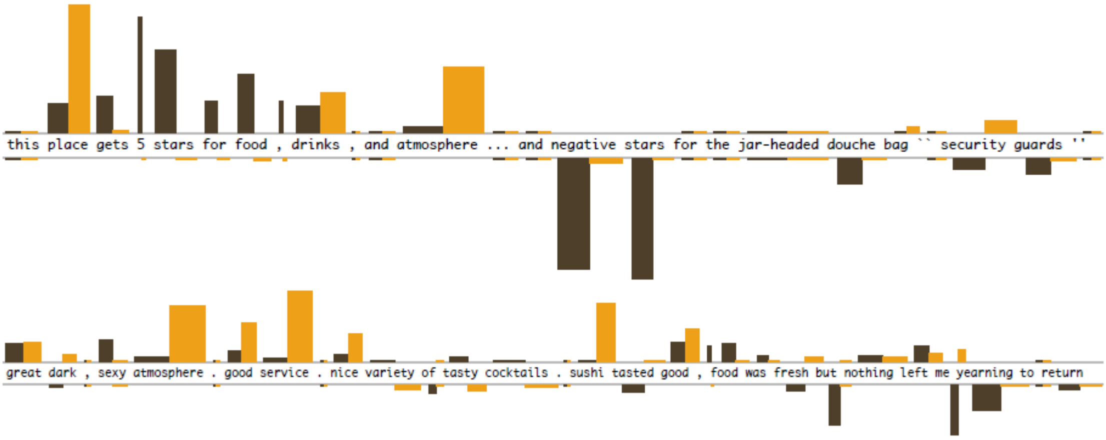

Figure 2.1: Local sentiment and price estimates for two test sentences (Chahuneau *et al.*, 2012).

task. Figure 2.1 shows the influence of each word, in a review sentence, on the predicted sentiment polarity (brown) and price range (yellow). The height of a bar is proportional to the sum of the feature weights for each unigram, bigram, and trigram feature containing the token. The first example presents a smooth change being expressed in terms of sentiment from the beginning to the end of the sentence. The second example is perhaps more difficult for sentiment analysis, since there are several positive words but the overall sentiment is essentially expressed in the final part. The model noted the constant positive sentiment at the beginning of the phrase, but also identified the fundamental negation, given the strong negative weight on bigrams such as *fresh but*, *left me*, and *me yearning*. In both examples, the yellow bars show that the price is reflected especially through isolated mentions of offerings and amenities, for instance through n-grams like *drinks, atmosphere, security*, and *good services*.

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%. The authors also mapped every word that appears in some review text according to the two-dimensions, i.e., sentiment and price. They observed that there were word groups with different characteristics, namely a group of words that appears in positive reviews of inexpensive restaurants, such as *very reasonable*, and a group of words that are used in negative reviews of more expensive restaurants, such as *no flavor*. These examples are represented in Figure 2.2.

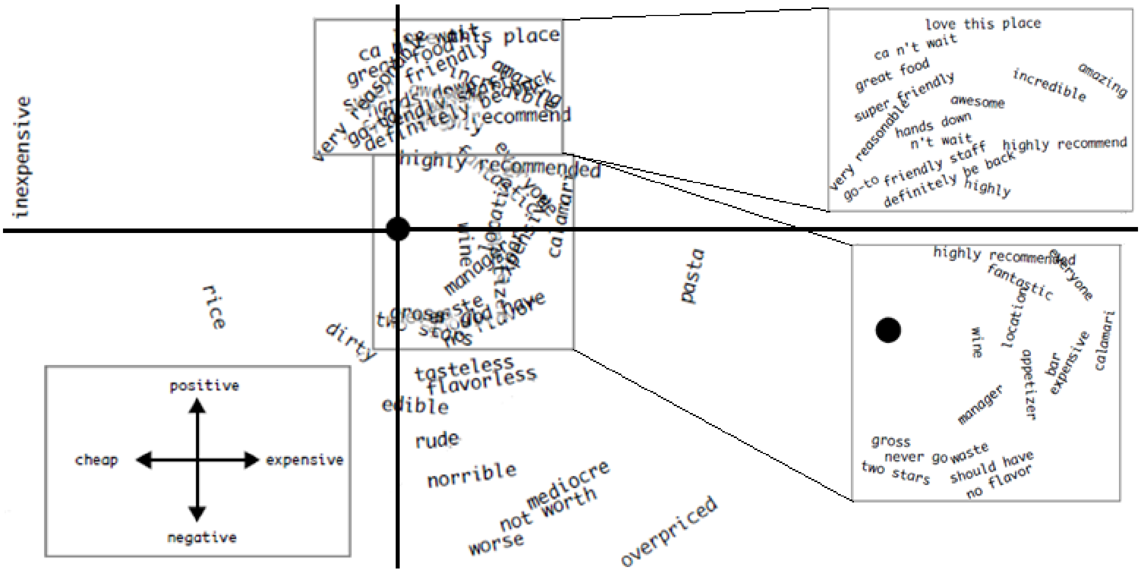

Figure 2.2: Top features for jointly predicting sentiment and price (Chahuneau *et al.*, 2012).

2.2 Textual Predictors of Congress Bill Survival

A U.S. Congressional bill is a textual artifact that must first pass through a series of hurdles to become a law, one of them being its consideration by a Congressional committee. In a related previous study Yano *et al.* (2012) evaluated predictive models to tell whether a bill will survive the congressional committee, starting with a strong baseline that uses features derived from the bill's sponsor and the committee it is referred to, and later augmenting the model with information from the textual contents of bills. The authors have experimentally shown that models using features derived from textual contents achieve a significant reduction in the prediction error, thus highlighting the importance of bill substance.

More specifically, Yano *et al.* proposed to leverage on the formalism of logistic regression models, considering l_1 regularization, to assign one of two classes to each bill, telling if it will survive or not. The authors collected the text of all bills introduced in the U.S. House of Representatives from the 103rd to the 111th Congresses (i.e., from 1/3/1993 to 1/3/2011), from the Library of Congress's Thomas website[1]. In this corpus, each bill is associated with its title, the bill's text, committee referral(s), and a binary value indicating whether or not the committee reported the bill to the chamber. Each bill is also associated to metadata properties, such as the sponsor's name. There were a total of 51,762 bills during the seventeen-year period that was considered, of which 6,828 bills survived the committee and progressed further. Bills from the 103rd to the 110th Congresses served as the training dataset, while the bills from the 111th Congress were

[1] `http://thomas.loc.gov/home/thomas.php`

used as the test dataset. The authors considered the following features for representing each bill, later augmenting the representation based on these features with textual information:

1. For each party p, a binary feature indicating if the bill's sponsor is affiliated or not with the specific party p.
2. A binary feature indicating if the bill's sponsor is affiliated in the same party as the committee chair where the bill was proposed.
3. A binary feature telling if the bill's sponsor is a member of the committee.
4. A binary feature indicating if the bill's sponsor is a majority member of the committee where the bill is being referred to.
5. A binary feature indicating if s the bill's sponsor is the chairman of the committee where the bill is being referred to.
6. For each House member j, a binary feature indicating if j is a sponsor of the bill.
7. For each House member j, a binary feature indicating if the bill is sponsored by j and referred to a committee that j chairs.
8. For each House member j, a binary feature indicating if the bill is sponsored by j and if j is in the same party as the committee chair.
9. For each state s, a binary feature indicating if the bill's sponsor is from s.
10. For each month m, a binary feature indicating if the bill was introduced during the specific month m.
11. For $v \in \{1, 2\}$, a binary feature indicating if the bill was introduced during the v-th year of the (two-year) Congress.

Regarding the textual features, the authors experimented with three different approaches:

1. Using unigram features from the body of each bill's text to pre-categorize bills into three generic classes (i.e., trivial, recurring, and important), later using these classes as features in the prediction model for bill success. For each of the three possible labels, the authors considered two classifiers trained with different hyperparameter settings, giving a total of 24 additional features.
2. Using the similarity towards past bills, in order to estimate votes by members of the committee on the bill. Using the cosine similarity between the TF-IDF vectors of each two bills,

the authors modeled the notion that representatives should vote on a bill x identically to how he voted on a similar bill x' through a simple probabilistic model, later quantizing the probability values into bins, and building 141 features with basis on these values;

3. Seeing the predictive model as a document classifier, and incorporating standard bag-of-words features directly into the model, rather than deriving functional categories or proxy votes from the text. The authors included unigram features from the body and unigram and bigram features for the title of the bill, resulting in a model with 28,246 features, of which 24,515 are lexical.

A most-frequent-class predictor (i.e., a constant prediction that no bill will survive the committee) achieves an error rate of 12.6%, whereas an l_1 regularized logistic regression model, using only the non-textual features, achieved an error rate of 11.8%. Unsurprisingly, the model's predictions are strongly influenced toward survival when a bill is sponsored by someone who is on the committee and/or in the majority party. On what regards the usage of textual features, Method 3 from the previous enumeration resulted in the best performance. Combined with baseline features, word and bigram features led to nearly 18% relative error reduction compared to the baseline, and 9% relative to the l_1 regularized logistic regression model. When using all three kinds of text features together, the authors report on an error reduction of only 2% relative to the bag of words model. Together, these results suggest that there is more information in the text contents than either on the functional categories or in the similarity towards past bills.

2.3 Characterizing Variation in Well-Being Using Tweets

Schwartz *et al.* (2013) reported on a study that analyzed the content of twitter messages from 1,300 different US counties, attempting to predict the subjective well-being of people living in these counties, as measured by overall life satisfaction (Diener, 2000; Diener & Suh, 1999; Diener *et al.*, 1985). Subjective well-being refers to how people evaluate their lives in terms of cognition (i.e., their general satisfaction with life) and emotion (i.e., positive and negative emotions) and, although several social media studies have focused on the analysis of the emotion component as expressed in the text (e.g., sentiment analysis applications (Pang & Lee, 2008)), the authors argue that most previous studies failed to capture the many nuances of subjective well being.

Specifically, the authors started by collecting approximately a billion tweets from June 2009 to March 2010, mapping these tweets to the corresponding US counties either through geospatial coordiantes associated to the tweets, or through the free-response location field that accompanies a tweeter message. After mapping the tweets and reducing the data to 1,300 different US counties, the authors were left with approximately 82 million tweets. Afterwards, the authors analyzed the language of these twitter messages through lexical and topical features, attempting to use language together with other popular predictors of county-level well being, such as the median age, gender (i.e., the percentage of female inhabitants), information on minorities (i.e., the percentage of black and hispanic inhabitants), the median household income, and educational attainment. The predictor variables besides language (i.e., besides the lexical and the topical features) drew on demographic and social-economic data from the US Census Bureau, and they were essentially used as controls (i.e., the authors attempted to see if predictive models based on language can add information beyond what these variables already contribute).

On what regards the lexical features, the authors relied on hand-built lists of words, including those from the psychological tool named Linguistic Inquire and Word Count (LIWC) (Pennebaker *et al.*, 2001), as well as terms associated to the PERMA (positive, emotion, engagement, relationships, meaning in life, and accomplishment) construct of well-being (Seligman, 2011). Each word in these lists is associated with a semantic category, such as positive emotion, leisure, engagement, etc. The actual features correspond to measurements of the percentage of a county's words within each given category (i.e., one feature for each category), using a log transformation to reduce the variance.

As for the topical features, the authors relied on clusters of lexico-semanticaly related words, derived automatically from a Latent Dirichlet Allocation topic model (Blei *et al.*, 2003). Specifically, the authors used an LDA model previously build by Schwartz *et al.* (2013) with basis on status updates from 18 million Facebook users. This LDA model considered 2000 different topics, and the authors argue that it captures well the different types of topics that are discussed within social media. The actual features correspond to the different topic probabilities for each county, obtained from the LDA model with basis on the concatenation of all twitter messages associated to a particular county. Again, a log transformation is used to reduce the variance on the features.

The actual predictive model corresponds to a Lasso linear regression model, using information from representative pooling surveys as the objective variable. A total of 75% of the tweets were used for model training (i.e., 970 counties), whereas the remaining 25% (i.e., 323 counties) were used for model validation. The quality of the results was measured through Pearson's correlation coefficient, computed between the predicted life satisfaction scores and those measured by the pooling surveys. Results showed that the control variables are more predictive than the LDA-based topic features alone, which in turn are more predictive than the lexicon-based features. However, combining the full set of features resulted in an increased performance, confirming that the words in tweeter messages contain information beyond that which is conveyed in the control variables that were considered for the study.

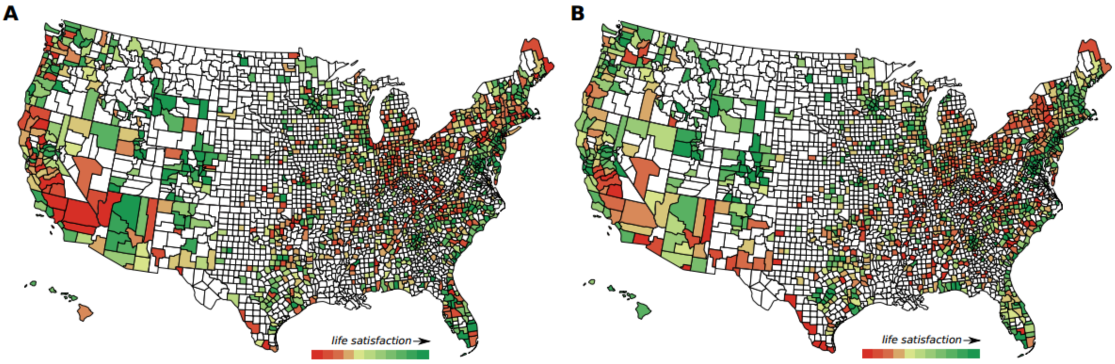

Figure 2.3: Map of the United States showing life satisfaction as measured using survey data and as predicted using the approach proposed by Schwartz *et al.* (2013).

Although the lexicon-based features added very little information over the topic-based features, the authors choose to keep them in their final predictive model, as they are useful for interpreting the results and for making comparisons against previous studies. Several LDA topics were highly predictive of positive well-being, including those related to outdoors (e.g., see/water, mountains or hiking) and recreation activities, as well as those related to learning and money. Topics associated to negative well being were less varied, involving the use of substantive words and words related to attitude (e.g., sick, hate, bored, etc.).

Figure 2.3 shows the regional variation in life satisfaction, as determined by survey data and as predicted by the text-driven approach proposed by the authors.

2.4 Predicting News Story Importance Using Their Text

Krestel & Mehta have reported on two studies that address the problem of anticipating a news story importance, (i.e., given a news item, automatically predicting if it will be of interest for a majority of users) (Krestel & Mehta, 2008, 2010). The authors argue that using natural language features for addressing this particular prediction task can have advantages, since user feedback, though very helpful for prediction (e.g., several previous work have reported on good results for this same prediction task, but relying on click data collected immediately after the publication of the news item (Szabo & Huberman, 2010; Tatar *et al.*, 2011; Yin *et al.*, 2012)), is associated with high latency, and sufficient user feedback may not be available when information is still new.

In a first experiment, the authors considered a learning setup whose objective was to predict if a news item lies in one of 4 categories (i.e., extremely important, highly important, moderately important, and unimportant), with basis on natural language features represented with a standard bag-of-words approach (Krestel & Mehta, 2008). In a second experiment, the authors used the Latent Dirichlet Allocation (LDA) topic model (Blei *et al.*, 2003) to find the latent factors behind important news stories, afterwards using these factors to train a classifier that enables them to see if new news items will become important in the future, as news item appear in services such as Google News (Krestel & Mehta, 2010).

We specifically have that, in the first experiment, the authors collected data from Google News[1], downloading stories displayed in the *World* category between November 15th 2007 and July 3rd 2008. This dataset covered a total of 1295 topics, each containing between 3 and 5 articles from the time when the topic first appeared. Google News uses a text clustering method to group similar news items, afterwards tracking updates and growth in interest with basis on these clusters. The authors argued that the relative importance of a particular news article is strongly tied to the cluster size of the corresponding topic, and so they started by assigning their data into different bins based on cluster size. Specifically, they used one bin for the topics with cluster size between 0 and 500, one for clusters 500 to 1000 articles, one for clusters 1000 to 2000 articles, and one bin for the news topics with more than 2000 articles in the corresponding cluster. These bins were assigned the values of one (i.e., unimportant), two (i.e., moderately important), three (i.e., highly important) and four (i.e., extremely important). The objective was to predict this value, with basis on the textual features.

The authors performed some cleaning on the HTML contents of the news pages, to remove navigation and advertisement information. The text of the pages was also processed in order to remove stopwords and to generate lemmas from the individual words. Each topic was represented using a space vector model, and the respective weights were computed using the TF-IDF scheme. Krestel & Mehta measured the accuracy of classification through of a 0-1 loss function over a test set. This function reports an error of zero if the correct label has been assigned, and of one otherwise. In the case of two class problems, the 0-1 function is very indicative of accuracy. However in the case of multi-class problems, this function considers a misclassification of highly important as unimportant to be the same as a classification as very important. Therefore, the authors also used the Mean Average Error (MAE) and the Root Mean Square Error (RMSE), using the label set $\{1,2,3,4\}$.

Regarding to the prediction accuracy, the results showed that textual features are indicative of importance, although the accuracy was not very high. The authors noticed that binary prediction, i.e., considering only the important and unimportant classes, achieved nearly 80% of accuracy,

[1] http://news.google.com

with linear SVMs. They also concluded that their classifiers are highly accurate in making prediction of truly important news correctly. For the task of predicting unimportant news, they achieved the highest precision using only nouns, whereas for the task of predicting important news, using all features resulted an increase of approximately 10% in terms of accuracy. Regarding to regression accuracy, we have that the regression task is more sensitive to larger misclassification errors. For the four-class problem, the best regression results were achieved by using all features. The lowest MAE was achieved when using only job titles. To find the most discriminative features, the authors analyzed the SVM model. They concluded that world leader's names are influential features, and disaster related words (e.g., *wreckage* and *explode*) also are highly influential. News that contain terms related to economics are usually also considered more important that others. Finally, Krestel & Mehta investigated if the weights of terms can change over time. They noticed that the changing of political environment had a influence over the importance for the names of politicians, e.g., *Abbas* had an importance peak in the month of May 2008, while the name *Musharraf* is loosing importance over time.

For the second experiment (Krestel & Mehta, 2010), the authors used again data collected from the Google News service, namely 3202 stories, gathering from 4 to 7 articles that were crawled from different sources, over a period of one year (i.e., the year of 2008). First, the authors explored the effectiveness of SVM based classifiers using term frequency vectors. This approach performed well, although it is difficult to generalize, due to the sparseness of features and redundancy. Therefore, the authors proposed to use LDA to identify latent factors derived from the text, using them as features for a classifier. In the context of the generative LDA model, the documents are represented as probabilist combinations of topics, denoted as $\mathrm{P}(z|d)$, with each topic being described by terms following another probability distribution, denoted by $P(w|z)$. We have that the LDA model can be formulated as follows:

$$\mathrm{P}(w_i) = \sum_{j=1}^{T} \mathrm{P}(w_i|z_i = j)\mathrm{P}(z_i = j) \tag{2.1}$$

In the formula, $\mathrm{P}(w_i)$ corresponds to the probability of word w_i appearing in a given document, and z_i refers to a latent topic. The term $\mathrm{P}(w_i|z_i = j)$ represents the probability of a word w_i within the topic j, while the term $\mathrm{P}(z_i = j)$ represents the probability of choosing a word form topic j in the document, and the parameter T corresponds to the number of topics, being used to adjust the degree of specialization of latent topics.

The authors compared the two approaches, and the results proved that the LDA approach yielded a better accuracy than the bag-of-words approach. The task of reducing the number of features, using LDA, improves the efficiency, interpretability, and accuracy. The authors concluded that the higher the number of latent topics, the more specific are the LDA topics. In terms of the correlation coefficient, when using the LDA approach, the authors measured a value of 0.47, whereas using the bag-of-words approach they achieved a score of 0.39.

2.5 Exploring Yelp Reviews in Forecasting Tasks

Yelp is an online service to help people find interesting local business (e.g., restaurants) that includes social networking features. On Yelp, the users can read restaurant reviews, and they can leave their reviews about the restaurants that they visited. Yelp leverages the wisdom of the crowd, with each business being reviewed many times. However, the higher the number of reviews left on the service, the more time-consuming and difficult it becomes for a consumer to process the underlying information. Several questions have been raised in the context of services such as Yelp, including the following:

1. Do online consumer reviews affect restaurant demand?
2. Can a group of amateur opinionators transform the restaurant industry, where heavily marketed chains and highly regarded professional critics have long had a stronghold?
3. Given a set of reviews in a service such as Yelp, what is the optimal way to construct an average rating for each local business?
4. Is it possible, given a particular review, to predict the number of users that will find the review useful?

To answer the first two questions, Luca (2011) combined a Yelp dataset with the revenues for every restaurant in Seattle (WA), between 2003 and 2009. In this work, the author examined the impact of Yelp reviews on revenues, specifically for chain restaurants. The study showed that a one-star increase on Yelp brings about to a 5 to 9 percent increase in revenue, but this effect only applies for independent restaurants. Moreover, the higher the penetration of Yelp, the more chain restaurants have declined in market share. The impact of online consumer reviews is larger for products of relatively unknown quality. Luca concludes that Yelp may help to drive worse restaurants out of business.

Regarding Question 3, Dai *et al.* (2012) developed a framework for better estimating business ratings, that allows reviewers to vary in stringency and accuracy, and to be influenced by existing reviews. This framework also takes into account that a restaurant's quality can change over time. The authors applied their approach to reviews from Yelp, to derive optimal ratings for each restaurant. To identify the relevant factors, the authors used variation in ratings within and across reviewers and restaurants. Dai *et al.* (2012) created optimal average ratings for all restaurants on Yelp, using their estimated parameters, and then compared them to the simple arithmetic mean displayed by Yelp. Regarding the results that were achieved, the authors found that the difference between the optimal average ratings and the simple average by Yelp is of more than 0.15 stars for 24-28% of the restaurants, and of more than 0.25 stars for 8-10% of the restaurants. In sum, a large gain in terms of rating accuracy can be acquired by implementing optimal ratings in a service such as Yelp.

An answer for the last question is presented by anonymous students of a machine learning course, taught by Nando de Freitas at the University of British Columbia, who have addressed the task of predicting Yelp review usefulness, with basis on the text[1]. The dataset collected from Yelp consisted of 229,907 reviews of 11,537 businesses, by 43,873 users. Yelp also provided a dataset that contains 8,282 sets of check-ins. The training data was gathered between March 2005 and January 2013. The data extraction was made by using text mining techniques, such as those implemented on the text mining package `tm`[2] for the `R` statistical programming language. The authors split the dataset into two parts, using the oldest 80% of the reviews to train the models, and the remaining 20% of the data to test the models. The authors used a simple bag-of-words approach to represent the text data in a document term matrix (DTM), in which the rows represent the documents, the columns refer to the words, and the entry in row i and column j represents the weighted frequency that the jth word appears in the ith document. The authors used Term Frequency-Inverse Document Frequency (TF-IDF) weighting, to calculate the importance of a word in the DTM.

Before extracting features from the text, the authors performed a data cleaning process, that includes removing text formatting, converting data into plain text, stemming words, removing whitespace and uppercase characters, and removing stopwords. To do this, they followed the workflow for preprocessing the data and extracting features, that is provided by the authors of the `tm` package. After this process, the final DTM contained 288 words of interest. The authors added to the DTM some useful features provided in supplementary datasets, creating some new features as well. Supplementary data allows connecting the reviews to the actual local businesses, personal information about the users, and check-ins data.

The authors modeled the response variable, corresponding to useful votes, as count data that takes positive integer values. The average number of votes in the dataset was of 1.387, and the data was highly skewed to the right, since over 40% of the 229,907 reviews received 0 votes.

[1] http://www.cs.ubc.ca/~nando/540-2013/projects/p9.pdf

[2] http://cran.r-project.org/web/packages/tm/index.html

Therefore, they transformed the discrete count data by applying the logarithm function to the response variable. The authors used a small constant smoothing value of $\varepsilon = 0.01$ to avoid the problem of computing the logarithm of 0. The authors tried fitting a non-parametric Random Forest regression model, on both the response variable and the log of the response. For the task of feature selection, they used these two models. Finally, the authors tried to fit fully parametric regression models, such as Negative Binomial (NB) and Zero Inflated Negative Binomial (ZINB), on the discrete response variable. The Random Forest and Lasso implementations were provided by the scikit-learn Python package.

Random Forest regression and the Lasso regression models assume that the response variable is unbounded, while in the Negative Binomial model, the response variable is considered as discrete and positive, and following a negative binomial distribution. To fit the fully parametrized NB regression, the authors used the `glm`[1] package available for `R`[2], with the variables selected by Lasso and Random Forest. The Zero Inflated Negative Binomial regression model is used to model count variables with excessive zeros, and it is usually effective for overdispersed count outcome variables. To amend the problem of excessive zeros on the Yelp review dataset, the authors also experimented with fitting a ZINB model. This assumes that the response comes from of a mixture of responses that correspond to zero votes with probability one, and responses that follow the negative binomial model. To fit the ZINB model, the authors used the `pscl`[3] package, also available for the `R` system for statistical computing.

Regarding to the evaluation of the different models, the authors used the Root Mean Squared Log Error (RMSLE), taking the logarithm of both the predicted and of the actual number of useful votes of reviews and measuring their differences.

The authors compared the results obtained from fitting the Random Forest and Lasso models using term frequency text features or TF-IDF weighted text features, on the log of the response, or on the discrete response. They concluded that, both in the Random Forest and Lasso models, using the TF-IDF weighted text features provides better predictions relative to the RMSLE metric. The effect of fitting the models on the log response shows that training Random Forests yielded better results, unlike Lasso that performed better on untransformed responses. In sum, the authors achieved the best results using Lasso on the TF-IDF weighted text features and over the discrete response variable. On what regards the parametric models, results showed that the ZINB regression model performed slightly better on the validation set. However, the parametric models performed worse than the Lasso and Random Forest regression models.

[1] http://cran.r-project.org/web/packages/glmnet/index.html

[2] http://www.r-project.org

[3] http://cran.r-project.org/web/packages/pscl/index.html

2.6 Summary and Critical Discussion

This chapter presented the most important related work previously developed in the area of text-driven forecasting, on which I based the developments made in the context of my MSc thesis. The most relevant group of related works is perhaps the one that corresponds to the studies made by Noah Smith and his colleagues at the Language Technology Institute from Carnegie Mellon University. In sum, we have that Smith (2010) proposed a generic approach for text-driven forecasting, with basis on fitting regression models that leverage features derived from the text contents, that are often noisy and sparse. He claimed that text-driven forecasting can be addressed through learning-based methodologies that are neutral with respect to different theories of language. Furthermore, the evaluation in this area can be objective, inexpensive, and theory-neutral, and thus text-driven forecasting tasks can support the effective comparison of different modelling choices for representing textual contents, in NLP applications.

All previous works that were presented in this chapter considered only the case of predicting values using English contents. In the context of my MSc thesis, particular emphasis is given to experiments with documents written in Portuguese.

Previous studies have also only considered relatively simple representations for the textual contents, for instance based on word features and TF-IDF weighting. In my work, I experimented with more sophisticated term weighting schemes, such as the Delta-TF-IDF and Delta-BM25 approaches, that measure the relative importance of a term in two distinct classes. In the next chapter, I will present the most important contributions of my MSc dissertation.

Chapter 3

Making Predictions with Textual Contents

In my study, similarly to Noah Smith and his colleges, I approached the problem of making predictions from textual contents as a regression task. This chapter details the developments made in the context of my MSc thesis, which built on and extended previous works in the area, e.g., by Noah Smith (2010), most of them performed with English contents. I specifically used texts written in Portuguese form different domains, and I used different representations for the words, different feature weighting schemes, and different learning methods.

The following section presents an overview on the considered methodology. Section 3.2 addresses the usage of features derived from cluster-based word representation, i.e., word clustering. Section 3.3 presents the approaches taken for representing textual contents as feature vectors. Finally, Section 3.4 presents the regression techniques that were considered, namely linear regression models and ensemble learning methods.

3.1 The General Approach

Text-driven forecasting concerns with the analysis of document collections, where each document corresponds to a character string, in a given natural language. Each document, in a given dataset, is made of simpler units, and in the context of tasks such as document classification and text-driven forecasting, documents are typically represented through sets or vectors derived such smaller units, like words, n-grams of words (i.e., sequences of n continuous words in a document) or n-grams of characters (i.e., sequences of n continuous characters in a document).

Considering these units for the computational representation, each document can be modeled as a vector of characteristics in a given vector space, in which the dimensionality corresponds to the number of different constituent elements (i.e., characteristics) that can be used in the formation of documents. 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 textual document thus 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. In the case of the experiments reported on this dissertation, the features are essentially the words that occur in the document collection, but in some of my experiments I 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.

In the general case, we have that the regression problem deals with the prediction of a response variable y given the values of a vector of predictor variables x. Considering $\mathcal{X}$ as the domain of x and $\mathcal{Y}$ as the domain of y, this problem can be reduced to a task of finding a function $\mathrm{d}(x)$, that maps each point in $\mathcal{X}$ to a point in $\mathcal{Y}$. The construction of $\mathrm{d}(x)$ requires the existence of a training set of n observations $L = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. For the case of regression problems, the criterion for choosing $\mathrm{d}(x)$ is usually the mean squared prediction error $\mathrm{E}\{\mathrm{d}(x) - \mathrm{E}(y|x)\}^2$, where $\mathrm{E}(y|x)$ is the expected value of y at x.

Over the years, many different learning methods have been proposed to address the task of finding the function $\mathrm{d}(x)$. In my research work, I focused on the usage of linear regression models with different types of regularization approaches, and regression approaches based on ensembles of trees. The problem of text-driven forecasting is therefore addressed as a regression task, according to the procedure shown in Figure 3.4. For each document in the set of training data, I built a representation by extracting the relevant features. Based on these representations, a regression model is trained and saved for latter application. For each document in the test set, the features are also extracted, in order to latter make predictions using the trained model. After predicting the target values for the test instances, the quality of the results is measured by comparing the predictions against the ground-truth values.

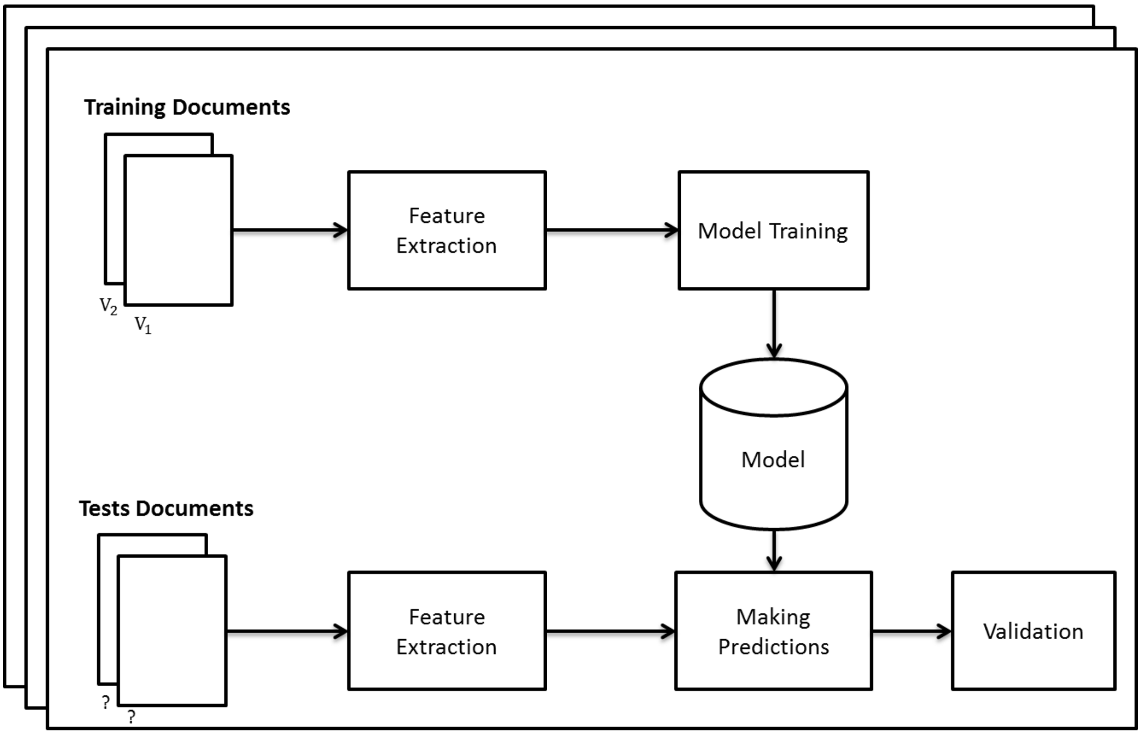

Figure 3.4: Text-driven forecasting as a regression task.

3.2 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, I 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 process is based on a language model leveraging bi-grams and classes, which is illustrated in Figure 3.5 and formalized bellow:

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

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

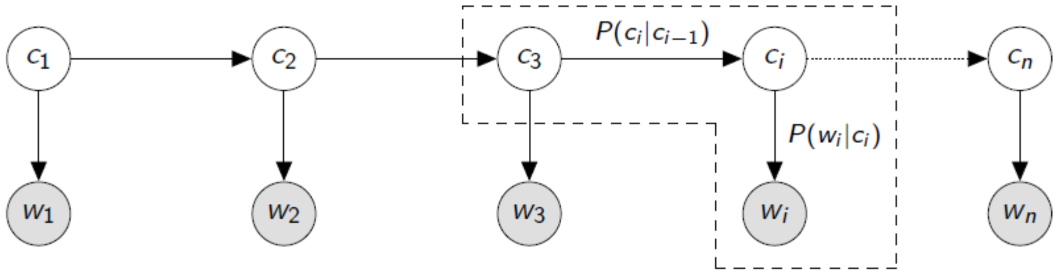

Figure 3.5: The class-based bi-gram language model, that supports the word clustering.

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 which results from maximizing the average mutual information between adjacent word clusters:

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

The estimation of the language model is 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 we have one cluster for each word), 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 we iteratively merge 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 $|C|$. Figure 3.6 shows an example of a binary tree resulting from Brown's clustering algorithm.

For this work, to induce the generalized representations of words, I used an open-source[1] 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*[2] newspaper, over a period of 10 years. We induced one thousand word groups, where each group has a unique identifier.

[1] `https://github.com/percyliang/brown-cluster`

[2] `http://www.publico.pt`

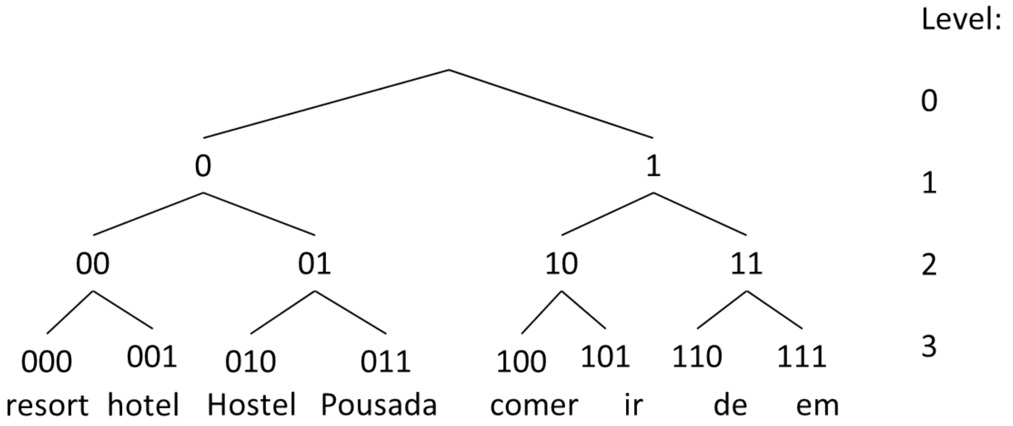

Figure 3.6: Example binary tree resulting from word clustering.

3.3 Feature Weighting

In this work, I also experimented with different ways to compute the feature weights to be used within my models, for both the textual terms and the word clusters. The feature weighting schemes that were considered include binary values, term frequency scores, TF-IDF, and also with more sophisticated term weighting schemes, such as the Delta-TF-IDF and Delta-BM25 schemes previously discussed by Martineau & Finin (2009) and by Paltoglou & Thelwall (2010).

In the case of binary weights, $w_{i,j}$ 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. Notice that the high frequency terms are more important for describing a document, which motivates the usage of the Term Frequency (TF) component. A variant of the TF weighting scheme that uses log normalization is given in Equation 3.2:

$$\mathrm{TF}_{i,j} = \begin{cases} \log_2(1 + frequency_{i,j}) & \text{if } frequency_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

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). Although there are multiple variants of the TF-IDF scheme, in all of them we have that the weight $w_{i,j}$ of a element i is proportional to the term frequency, i.e., the more often the element i appears in document j, the higher its weight $w_{i,j}$, and inversely proportional to the document frequency of element i.

The inverse document frequency is a measure of element importance within the collection of documents. An element that appears in most of the documents of a given collection is not important to discriminate between the different documents. So, the IDF is the inverse of the number of times that element i occurs in all documents, typically considering a base 2 logarithmic decay as shown in the equation bellow:

$$\mathrm{IDF}_i = \log_2\left(\frac{N}{n_i}\right) \tag{3.3}$$

In the formula, N is the total number of documents in the collection, and n_i is the number of documents containing element i. The TF-IDF weight of an element i for a document j is thus given by the combination of Formulas 3.2 and 3.3:

$$\mathrm{TF\text{–}IDF}_{i,j} = \begin{cases} \log_2(1 + frequency_{i,j}) \times \log_2\left(\frac{N}{n_i}\right) & \text{if } n_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The $\mathrm{Delta\text{–}TF\text{–}IDF}$ and $\mathrm{Delta\text{–}BM25}$ schemes measure the relative importance of a term in two distinct classes. In the context of my regression problems, I 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 value greater or equal to the median target value in the training data, and those that are less or equal than the median value.

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

$$\Delta\mathrm{TF\text{–}IDF}_{i,j} = \mathrm{TF}_{i,j} \times \log_2\left(\frac{N_{pos}}{n_{i,pos}}\right) - \mathrm{TF}_{i,j} \times \log_2\left(\frac{N_{neg}}{n_{i,neg}}\right) = \mathrm{TF}_{i,j} \times \log_2\left(\frac{N_{pos} \times n_{i,neg}}{n_{i,pos} \times N_{neg}}\right)$$

In the formula, each N_c corresponds to the number of training documents respectively in each 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 $\mathrm{Delta\text{–}TF\text{–}IDF}$ performs significantly better than the traditional $\mathrm{TF\text{–}IDF}$ or the binary weighting schemes (Martineau & Finin, 2009).

Paltoglou & 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 $\mathrm{Delta{-}BM25}$ 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:

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

3.4 Regression Models

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

3.4.1 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

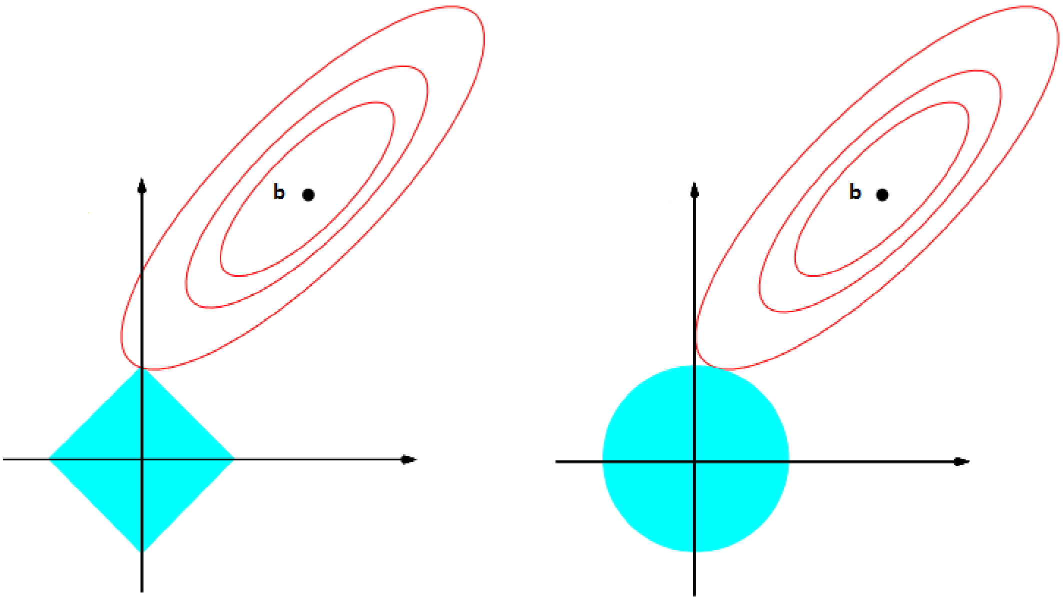

Figure 3.7: Estimating Lasso and Ridge regression.

$\lambda = 0$ 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 according to the following optimization problem:

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

Figure 3.7 illustrates a geometrical interpretation for the Lasso (left) and the Ridge (right) regularization approaches, when there are only two parameters. The elliptical contours correspond to the least squares error function. The constraint region corresponding to the Lasso is the blue diamond, and for Ridge regression is the blue disk. Both models find the first point where the elliptical contours reach the constraint region.

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 the aforementioned limitations Zou & 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 & Kim, 2006), or other convex optimization methods based on iterative computations (Boyd & Vandenberghe, 2004), such as SpaRSA (Wright *et al.*, 2009). In this study, I used the implementation available from the scikit-learn Python machine learning package.

3.4.2 Ensemble Learning Methods

Regression trees are another type of machine-learning method for constructing prediction models from data. A regression tree is essentially a tree-structured solution to regression problems in which a constant, or a relatively simple regression model, is fitted to the data in each partition. Regression trees are typically trained using fast divide and conquer greedy algorithms that recursively partition the given training data into smaller subsets.

A related idea is that of ensemble learning methods, which attempt to combine multiple models, usually relatively simple models based on trees, to achieve better predictive performance (Sewell, 2011). In my experiments, I used two types of ensemble regression methods, namely Random Forests, and Gradient Boosted Regression Trees, again as implemented within the scikit-learn Python machine learning 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 (see Figure 3.8), 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

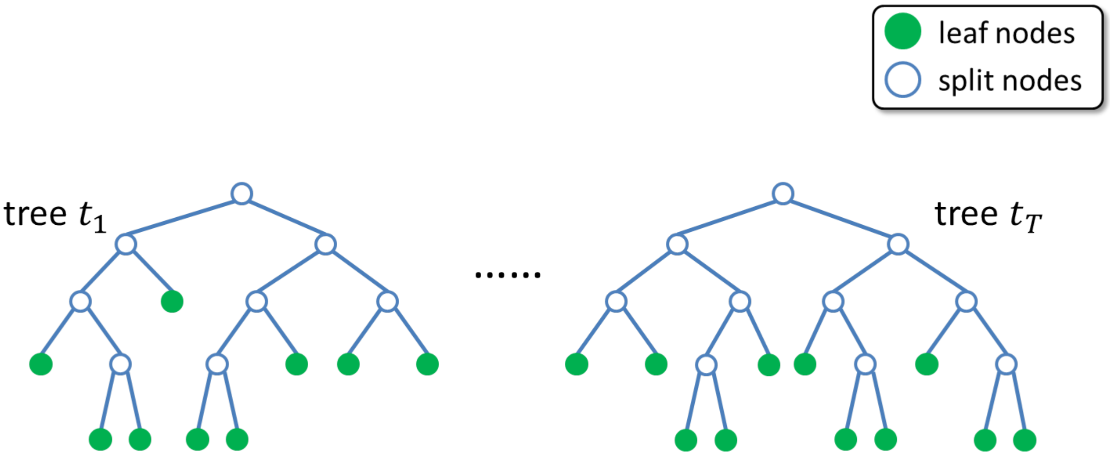

Figure 3.8: Multiple regression trees used on the Random Forest method.

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 ensemble.

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 that we are trying to minimize:

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

The optimization problem is typically addressed numerically via a steepest descent procedure (Boyd & Vandenberghe, 2004).

3.5 Summary

Ths chapter described the proposed regression-based methodology for addressing text-driven forecasting tasks. I discussed the approaches used for representing textual contents as feature vectors, as well as the schemes that can be used to compute the feature weights (e.g., binary weights, term frequency, the TF-IDF weighting scheme, and more sophisticated term weighting schemes, such as the Delta-TF-IDF and Delta-BM25). Finally, I presented several types of regression models that can support text-driven forecasting tasks. I focused on linear regression models with different regularization approaches, and on models based on ensembles of trees.

In the next chapter, I will present an experimental comparison for the different techniques that were discussed here, with basis on documents written in Portuguese from three distinct domains, namely (i) textual descriptions from Lifecooler, for hotels and restaurants, and (ii) movie reviews from Portal do Cinema, together with metadata from Instituto do Cinema e do Audiovisual.

Chapter 4

Experimental Validation

This chapter presents a set of validation experiments in which different modeling approaches for text-driven forecasting tasks were compared. In the following section, I present the datasets associated to each of the experimental domains that were considered, and the evaluation methodologies that I used. Then, I discuss the results achieved in the different experiments, comparing different types of regression models, different feature weighting schemes, and different sets of features. The chapter ends with an overview on the main conclusions.

4.1 Datasets and Methodology

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, I crawled textual descriptions from Lifecooler[1]. 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, I used review data from Portal do Cinema[2], and metadata from Instituto do Cinema e do Audiovisual[3], 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

[1] http://www.lifecooler.com

[2] http://www.portal-cinema.com

[3] http://www.ica-ip.pt

Table 4.1: 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 document	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 in target values	93.42	51.67	8.41	229.21

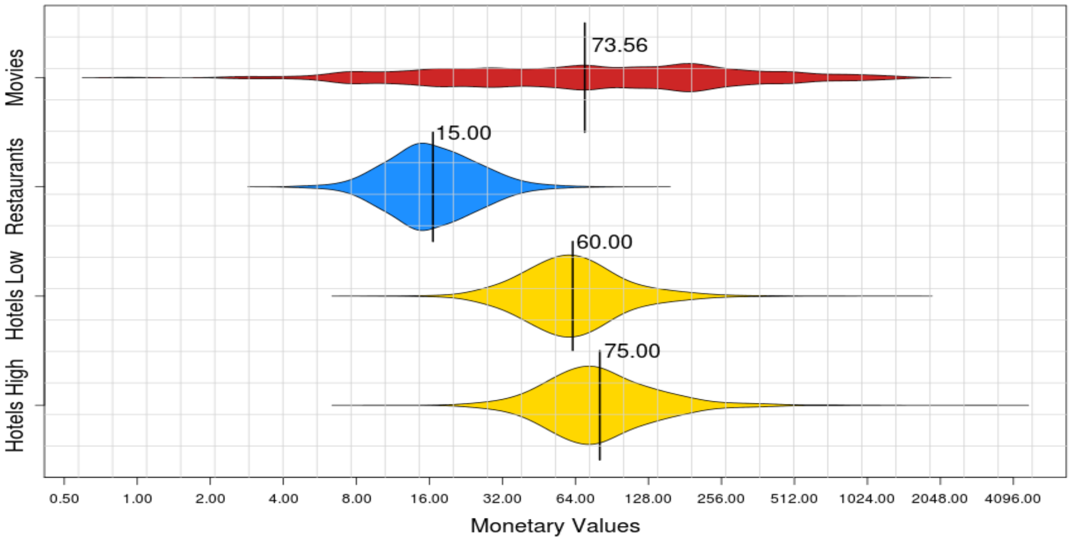

Figure 4.9: Distribution for the target values, in the hotels, restaurants, and movies datasets.

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 Web sites were included in my dataset, and I ended up with a set of 502 movies. The main characteristics of all three data datasets are presented in Table 4.1. 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 three 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 4.9.

All the available text was used in my experiments (e.g., for hotels, I used the hotel name and textual description, while for restaurants I used the restaurant name, the textual description, the specialities, and the menus). For movies, I used the movie name, and the textual review). In

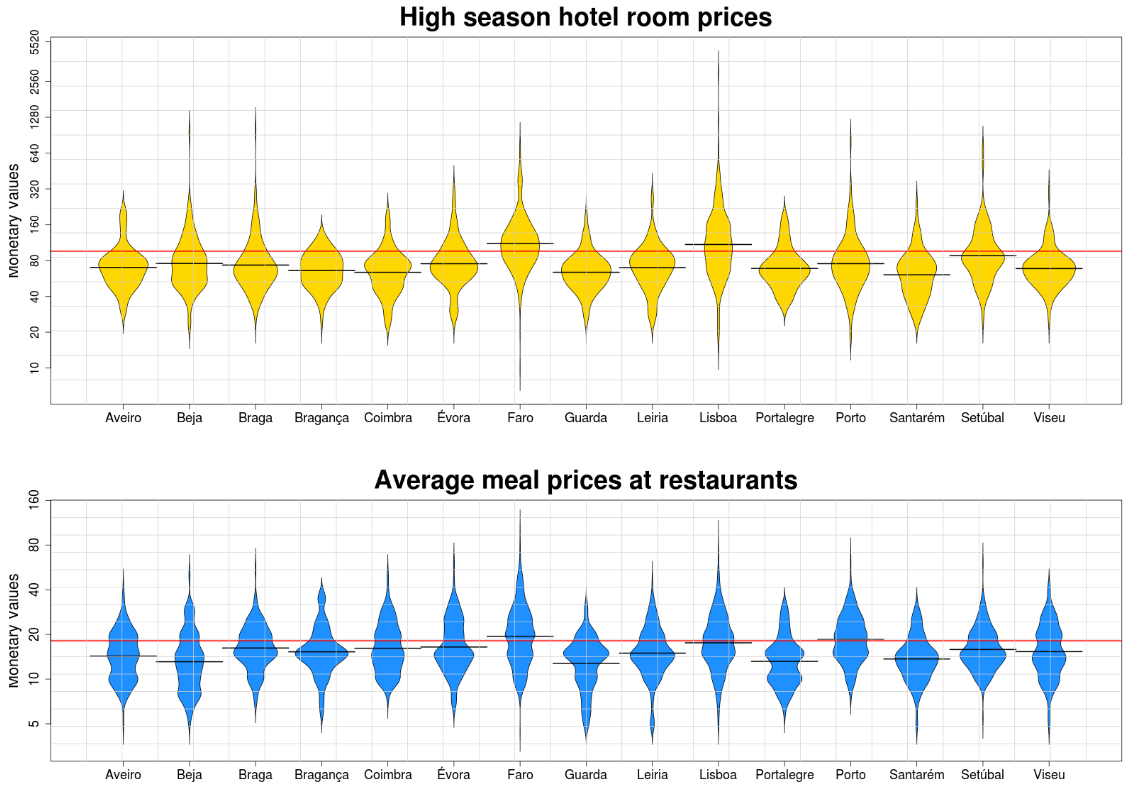

Figure 4.10: Distribution for the target values, per district in Continental Portugal.

addition to textual features, in some of the experiments I used location features (i.e., the administrative districts), for hotels and restaurants, type features for restaurants, or the number of screens that the movie was played at. The location features (i.e., administrative districts) can naturally influence how expensive is a hotel or a restaurant. For instance, as one can see in Figure 4.10, 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.

All my 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 4.5. 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 4.6.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (4.5) \qquad\qquad \mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (4.6)$$

Table 4.2: 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

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.

The MAE is perhaps the most natural and unambiguous measure of average error magnitude. On the other hand, the RMSE is one of the most widely reported error measures in the literature, and has also been widely used for comparing forecasting methods. This metric is more sensitive to the occasional large error.

4.2 Experimental Results

In a first experiment, I 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, I compared my 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 one can see in Table 4.2, regression models using the text achieve better results than the considered baselines. 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, I 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, I inspected the feature weights (i.e., the regression coefficients) of my learned models, averaging the weights of each

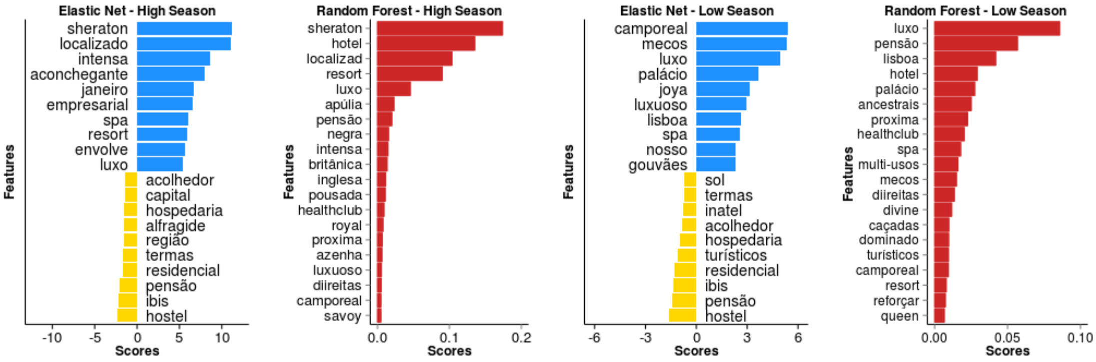

Figure 4.11: The 20 most important features in the case of predicting hotel room prices.

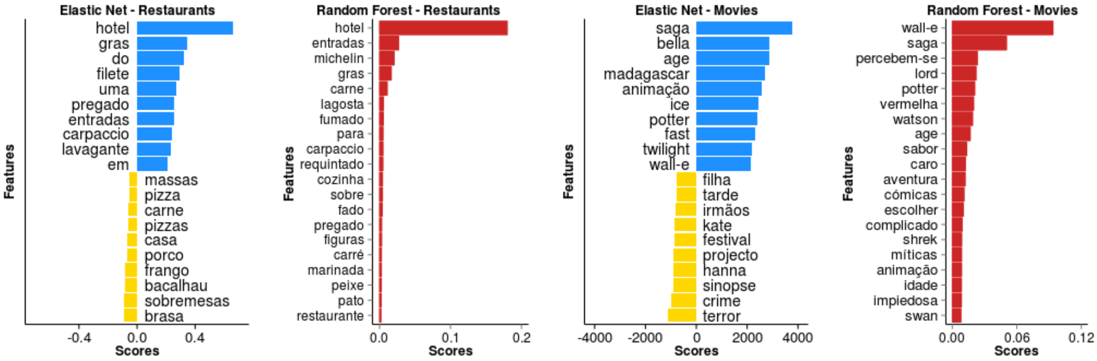

Figure 4.12: The 20 most important features in the case of predicting meal prices at restaurants, or in the case of predicting movie box-office results.

feature over the multiple folds of my 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 tree 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 my cross-validation experiments, one can estimate a feature importance score.

Figure 4.11 plots the 20 most important features in terms of either the linear regression coefficients (i.e., 10 with the highest positive value, and 10 with highest negative value), 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 *hotel* seem to indicate higher prices, whereas terms such as *pensão* or *camporeal* are associated with lower prices.

Table 4.3: 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**

Table 4.4: 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 4.12 also plots the 10 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*, *animação* or *terror* seem to provide goog clues for estimating the target values.

Tables 4.3 and 4.4 show a comparison of the results obtained with the best models, i.e., linear regression with Elastic Net regularization and Random Forest regression, respectively, using each of the representations for the textual contents that were described in Chapter 3. The richer document representation is perhaps Delta-BM25, although Delta-TF-IDF or TF-IDF alone achieved very similar results, particularly on the case of models based on Random Forest approach.

In addition to the textual contents, I 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 a set of 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 an on for the dimensions the last position on the feature vector.

Table 4.5: 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 4.6: 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 4.5 lists the corresponding results when predicting hotel room prices with different feature sets. The combination of the textual contents, the metada, 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 4.6 shows the results for predicting average meal prices for restaurants, using different feature sets. With the Elastic Net method, the experiment 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 4.7 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 4.13.

4.3 Summary

This chapter described a set of experiments validating the techniques proposed in my MSc dissertation. I specifically addressed the usage of regression models for text-driven forecasting

Table 4.7: 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

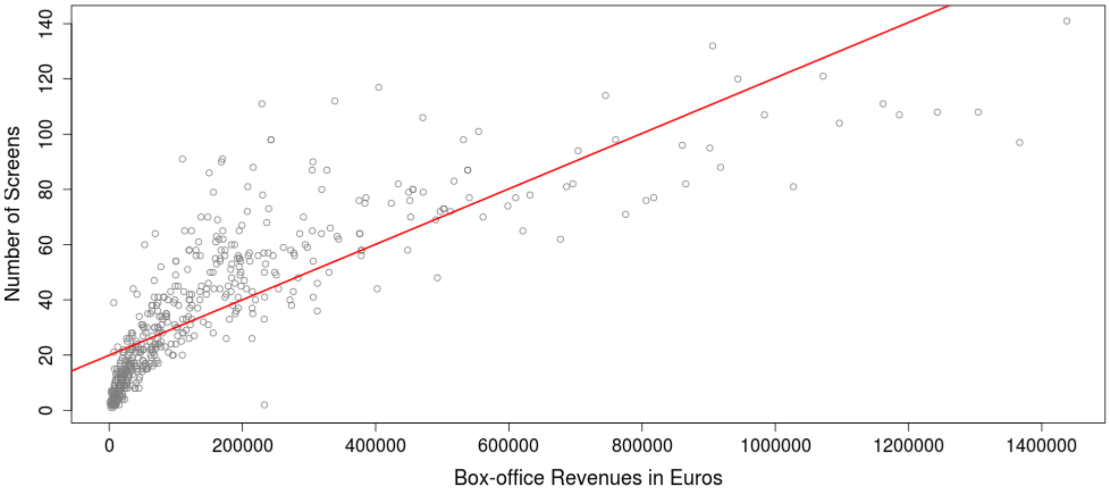

Figure 4.13: Box-Office revenues versus the number of screens on which the movie was shown.

Table 4.8: 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**

tasks, and I considered Portuguese texts from three different domains, collected from Lifecooler and from Portal do Cinema. I experimented with different types of regression models, with state-of-the-art feature weighting schemes, and with different sets of features.

On what regards hotels and restaurants, I compared the results of prediction models using textual information against baselines such as predicting the training set average value, or the average value per location. In the case of movies, I also experimented with using just the number of screens, a variable that is highly correlated with box-office results. An overview on the obtained results is reported in Table 4.8. In sum we have that better results were achieved when using the average value per location, when compared to using the average over the entire training dataset (i.e., a difference of almost 4 Euros, in the case of high season hotel prices). We can also conclude that regression models using textual contents indeed result in gains for the prediction accuracy over the considered forecasting tasks.

Chapter 5

Conclusions and Future Work

This dissertation presented an experimental study in which I tried to make predictions with textual contents written in Portuguese, using documents from three distinct domains and using different types of regression models. The specific tasks considered in my MSc thesis involved 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 named Lifecooler;
- Average meal prices in Portuguese restaurants, using textual descriptions for the restaurants and their menus, as collected from the same Web portal from the previous item;
- 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 site named Portal do Cinema.

I explored the usage of different types of regression models, with state-of-the-art feature weighting schemes, and with features derived from cluster-based word representations. The results clearly shown that prediction models using the textual information achieve better results than baselines such as the average value for the target variable. I also concluded that using richer document representations, involving Brown clusters and the Delta-TF-IDF feature weighting scheme, may result in slight performance improvements.

5.1 Main Contributions

The main contributions resulting from this work can be summarized as follows:

- I developed a software framework to support the experiments made in the context of my MSc thesis. This software framework includes programs for collecting data from the Web and for data processing, as well as methods for computing different feature weighting schemes, and for feature analysis. This framework can now be used in other research studies in text-driven forecasting.

- Previous works only considered the case of predicting some extrinsic real-word values using English contents, whereas this thesis gave particular emphasis to experiments with contents written in Portuguese. The results proved that it is possible to address different types of forecasting tasks using texts written in Portuguese. For all three domains that were considered, the models that used features derived from textual contents outperformed baselines such as the average value.

- Unlike previous works that focused only in the usage of linear regression models, in the context of my work, I compared linear regression models, using different model regularization approaches, against models based on ensembles of trees, such as Random Forests and Gradient Boosted Regression Trees. The best results were achieved with linear regression using Elastic Net regularization. The Random Forest method reached slightly lower results, and this was the best model in case of ensemble learning methods.

- I experimented with the usage of features derived from cluster-based word representations. I specifically used an open-source implementation of Brown's word clustering algorithm. In order to induce the representations of words, I used Portuguese texts corresponding to a very large set of phrases that combines the CINTIL corpus of modern Portuguese, with news articles published in the *Público* newspaper. In some of the experiments, the combination of the textual terms and the word clusters achieved slightly better results, compared against models that used only textual terms.

- I compared the results obtained with the state-of-the-art feature weighting schemes, such as Delta-BM25 or Delta-TF-IDF, with more traditional feature weighting schemes, such as binary weights, the term frequency, and TF-IDF. The richer representation is perhaps Delta-BM25, although the obtained results with Delta-TF-IDF or TF-IDF alone are almost similar.

5.2 Future Work

Despite the interesting results, there are also many ideas for future work. Given that I only had access to relatively small training datasets, I believe that one interesting path for future work concerns with evaluating semi-supervised learning techniques, capable of levering large amounts of non-labeled data for text driven forecasting (Chapelle *et al.*, 2006).

It also seems reasonable to assume that the cues to correctly estimating a given target value, with basis on a textual document, may lie in a handful of the document's sentences. Yogatama & Smith (2014) have, for instance, introduced a learning algorithm that exploits this intuition through 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, I would also like to experiment with other types of representations based on distributional similarity, such as the word embeddings proposed by Mikolov *et al.* (2013).

Bibliography

ARMSTRONG, J.S. & COLLOPY, F. (1987). Regression Methods for Poisson Process Data. *Journal of the American Statistical Association*, **82**.

BARRETO, F., BRANCO, A., FERREIRA, E., MENDES, A., FERN, M., NASCIMENTO, A.B.D., NUNES, F. & SILVA, J.R. (2006). Open resources and tools for the shallow processing of portuguese: The TagShare project. In *Proceedings of the International Conference on Language Resources and Evaluation*.

BLEI, D.M., NG, A.Y. & JORDAN, M.I. (2003). Latent Dirichlet Allocation. *Machine Learning Research*, **3**.

BOLLEN, J., MAO, H. & ZENG, X.J. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, **1**.

BOYD, S. & VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge University Press.

BREIMAN, L. (1996). Bagging Predictors. *Machine Learning*, **24**.

BREIMAN, L. (2001). Random Forests. *Machine Learning*, **45**.

BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A. & STONE, C.J. (1984). Classification and regression trees. Chapman and Hall/CRC.

BRITO, I. & MARTINS, B. (2014). Making predictions with textual contents: Experiments with portuguese documents from three distinct domains. *Springer*, **1**.

BROWN, P.F., DE SOUZA, P.V., MERCER, R.L., PIETRA, V.J.D. & LAI, J.C. (1992). Class-based *n*-gram models of natural language. *Computational Linguistics*, **18**.

CHAHUNEAU, V., GIMPEL, K., ROUTLEDGE, B.R., SCHERLIS, L. & SMITH, N.A. (2012). Word salad: Relating food prices and descriptions. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*.

CHAPELLE, O., SCHÖLKOPF, B. & ZIEN, A. (2006). *Semi-supervised learning*. MIT Press.

DAHLLÖF, M. (2012). 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**.

DAI, W., JIN, G.Z., LEE, J. & LUCA, M. (2012). Optimal aggregation of consumer ratings: An application to Yelp.com. *The National Bureau of Economic Research*, **13**.

DIENER, E. (2000). Subjective well-being: The science of happiness and a proposal for a national index. *American Psychology*, **55**.

DIENER, E. & SUH, E. (1999). Subjective well-being and age: An international analysis. In *Proceedings of the Annual Review of Gerontology and Geriatrics*.

DIENER, E., EMMONS, R.A., LARSEN, R.J. & GRIFFIN, S. (1985). The satisfaction with life scale. *Personality Assessment*, **46**.

DRUCKER, H., BURGES, C.J.C., KAUFMAN, L., SMOLA, A. & VAPNIK, V. (1997). Support vector regression machines. In *Proceedings of the Conference on Neural Information Processing Systems*.

FRIDMAN, J., HASTIE, T. & TIBSHIRANI, R. (2008). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**.

FRIEDMAN, J.H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, **29**.

HONG, Y. & SKIENA, S. (2010). The wisdom of bookies? sentiment analysis vs. the NFL point spread. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*.

JOSHI, M., DAS, D., GIMPEL, K. & SMITH, N.A. (2010). 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.*

KIM, Y. & KIM, J. (2006). Gradient Lasso for feature selection. In *Proceedings of the International Conference on Machine Learning*.

KOGAN, S., LEVIN, D., ROUTLEDGE, B.R., SAGI, J.S. & SMITH, N.A. (2009). Predicting risk from financial reports with regression. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

KRESTEL, R. & MEHTA, B. (2008). Predicting news story importance using language features. In *Proceedings of the IEEE Conference of Web Intelligence and Intelligent Agent Technology*.

KRESTEL, R. & MEHTA, B. (2010). Learning the importance of latent topics to discover highly influential news items. In *Proceedings of the Annual German Conference on Artificial Intelligence*.

LERMAN, K., GILDER, A., DREDZE, M. & PEREIRA, F. (2008). Reading the markets: forecasting public opinion of political candidates by news analysis. In *Proceedings of the International Conference on Computational Linguistics*.

LUCA, M. (2011). Reviews, reputation, and revenue: The case of Yelp.com. *Harvard Business School*, **12**.

LUO, X., ZHANG, J. & DUAN, W. (2013). Social media and firm equity value. *Information Systems Research*, **24**.

MARTINEAU, J. & FININ, T. (2009). Delta TF-IDF: An improved feature space for sentiment analysis. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*.

MCCULLAGH, P. (1980). Regression models for ordinal data. *Journal of Royal Statistical society*, **42**.

MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. & DEAN, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Neural Information Processing Systems*.

MITCHELL, L., FRANK, M.R., HARRIS, K.D., DODDS, P.S. & DANFORTH, C.M. (2013). The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place. *PLOS ONE*, **8**.

O'CONNORY, B., BALASUBRAMANYANY, R., ROUTLEDGE, B.R. & SMITH, N.A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*.

PALTOGLOU, G. & THELWALL, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

PANG, B. & LEE, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, **2**.

PENNEBAKER, J.W., BOOTH, R.J. & FRANCIS, M.E. (2001). *Linguistic Inquiry and Word Count (LIWC): A Computerized Text Analysis Program*. Lawrence Erlbaum Associates.

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

SCHUMAKER, R.P. & CHEN, H. (2009). Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems*, **27**.

SCHWARTZ, H.A., EICHSTAEDT, J.C., KERN, M.L., DZIURZYNSKI, L., PARK, M.A.G.J., LAKSHMIKANTH, S.K., JHA, S., SELIGMAN, M.E.P., UNGAR, L. & LUCAS, R.E. (2013). Characterizing geographic variation in well-being using tweets. In *Proceeding of the AAAI International Conference on Weblogs and Social Media*.

SELIGMAN, M. (2011). *Flourish: A Visionary New Understanding of Happiness and Well-Being*. Free Press.

SEWELL, M. (2011). Ensemble methods. Tech. Rep. RN/11/02, University College London Departament of Computer Science.

SMITH, N.A. (2010). Text-Driven Forecasting.

SZABO, G. & HUBERMAN, B.A. (2010). Predicting the popularity of online content. *Communications of the ACM*, **53**.

TATAR, A., LEGUAY, J., ANTONIADIS, P., LIMBOURG, A., DE AMORIM, M.D. & FDIDA, S. (2011). Predicting the popularity of online articles based on user comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*.

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

TIRUNILLAI, S. & TELLIS, G.J. (2012). Does chatter really matter? Dynamics of User-Generated Content and Stock Performance. *Information Systems Research*, **31**.

TURIAN, J., RATINOV, L. & BENGIO, Y. (2010). Word representation: a simple and general method for semi-supervised learning. In *Proceedings of the Annual Meeting of the Association for Computional Linguistics*.

WRIGHT, S.J., NOWAK, R.D. & FIGUEIREDO, M.A.T. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, **57**.

YANO, T. & SMITH, N.A. (2010). What's worthy of comment? Content and Comment Volume in Political Blogs. In *Proceedings of the AAAI International Conference on Weblogs and Social Media*.

YANO, T., COHEN, W.W. & SMITH, N.A. (2009). 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*.

YANO, T., SMITH, N.A. & WILKERSON, J.D. (2012). Textual predictors of bill survival in congressional committees. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

YIN, P., LUO, P., WANG, M. & LEEA, W.C. (2012). A straw shows which way the wind blows: ranking potentially popular items from early votes. In *Proceedings of the ACM International Conference on Web Search and Data Mining*.

YOGATAMA, D. & SMITH, N.A. (2014). Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proceedings of the International Conference on Machine Learning*.

ZOU, H. & HASTIE, T. (2005). Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society Series B*, **67**.