INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

# Text Mining Methods for Mapping Opinions from Georeferenced Documents

**Duarte Choon Dias**

Dissertation submitted to obtain the Master Degree in

**Information Systems and Computer Engineering**

**Jury**

| | |
|---|---|
| President: | Prof. Dr. António Manuel Ferreira Rito Silva |
| Advisor: | Prof. Dr. Bruno Emanuel da Graça Martins |
| Members: | Prof. Dra. Maria Luísa Torres Ribeiro Marques da Silva Coheur |

**October 2012**

# Abstract

With the growing availability of large volumes of textual information on the Web, text mining techniques have been gaining a growing interest. One specific text mining problem that is increasingly relevant relates to the detection of textual expressions that refer to opinions on certain topics and services. A second text mining problem, which has also been gaining a growing interest, is the identification of the geographic location that best relates to the contents of particular documents. In my MSc thesis, I empirically compared automated techniques, based on language models, for assigning documents to opinion classes and to the geospatial coordinates of latitude and longitude that best summarize their contents. Using this information, I then analyzed the possibility of building thematic maps portraying the incidence of particular classes of opinions, as extracted from documents, in different geographic areas. An extensive experimental validation has been carried out over the different components, using documents from Wikipedia and reviews from Yelp. The best performing method for geocoding textual documents combines character-based language models with a post-processing technique that uses the coordinates from the 5 most similar training documents, obtaining an average prediction error of 265 Kilometers, and a median prediction error of just 22 Kilometers. In what concerns opinion mining, analysis of opinion was done in a two-point scale schema (i.e., polarity of opinion) and a five-point scale schema (i.e., considering five degrees of opinions). The best performing methods used character-based language models, and for which the two-point scale case achieved an accuracy of 0.80. The best performing method for the five-point scale, based on a hierarchical classifier, achieved an accuracy of 0.50. A technique known as Kernel Density Estimation was used in the development of the thematic maps, and an empirical analysis has shown that the maps obtained through automatic extraction indeed correspond to an accurate representation for the geographic distribution of opinions.

**Keywords:** Geographic Information Retrieval , Opinion Mining , Thematic Mapping

# Resumo

Com a crescente disponibilidade de grandes volumes de informação textual na Internet, técnicas de *text mining* têm vindo a ganhar um interesse crescente. Um dos problemas específicos de *text mining* diz respeito à detecção de expressões textuais que se referem a opiniões sobre determinados temas e serviços. Um segundo problema de *text mining*, que tem vindo a ganhar atenção por parte da comunidade científica, é a identificação da localização geográfica que melhor se relaciona com o conteúdo de documentos. Na minha tese de mestrado, eu comparei empiricamente técnicas automatizadas, baseadas em modelos de linguagem, para a atribuição de documentos a classes de opinião e às coordenadas geoespaciais de latitude e longitude que melhor resumem os seus conteúdos. Usando essa informação, analisei a possibilidade de construção de mapas temáticos que retratam a incidência de determinadas classes de opiniões, extraídos de documentos, em diferentes áreas geográficas. Uma extensa validação experimental foi realizada ao longo dos diferentes componentes, usando documentos da Wikipédia e opiniões do website Yelp. O melhor método desempenho para a geocodificação de documentos textuais combina modelos de linguagem baseados em caracteres com uma técnica de pós-processamento que usa as coordenadas dos cinco documentos de treino mais similares, obtendo um erro de previsão médio de 265 quilómetros, e um erro de previsão mediano de apenas 22 quilómetros. No que diz respeito ao *mining* de opiniões, a análise de opiniões foi feita em um esquema de escala de dois pontos (ou seja, a polaridade de opinião) e um esquema de escala de cinco pontos. Os melhores métodos de desempenho utilizam modelos de linguagem baseados em caracteres. A escala de dois pontos obteve uma exatidão de 0,80, enquanto que o melhor método de desempenho para a escala de cinco pontos, baseado num classificador hierárquico, obteve uma exatidão de 0,50. Utilizou-se uma técnica conhecida como estimativa da densidade Kernel para o desenvolvimento de mapas temáticos, e uma análise empírica mostrou que os mapas obtidos através de extracção automática de facto correspondem a uma representação precisa para a distribuição geográfica de opiniões.

**Keywords:** Extracção de Informação Geografica , Opinion Mining , Mapas Temáticos

# Acknowledgements

Tenho muito a agradecer a várias pessoas pela ajuda imprescindível ao longo do tempo em que trabalhei na minha tese de mestrado. Gostaria de começar por agradecer ao meu orientador Professor Bruno Martins, e ao meu co-orientador Ivo Anastácio pela sua ajuda, apoio e inacreditável disponibilidade, sem os quais esta fase final de curso não teria sido possível.

Gostaria ainda de agradecer o suporte financeiro da Fundação para a Ciência e Tecnologia (FCT), através da bolsa do projecto SinteliGIS.

Um agradecimento especial aos meus amigos *tagusianos*, pelo espirito de camaradagem e por estarem sempre presentes para todos os momentos desta aventura.

Por último, gostaria de estender os meus agradecimentos a todos aqueles, que embora não tenham feito parte da minha vida académica, tiveram um especial contributo. Gostaria assim de agradecer aos meus pais, por me terem ensinado que tudo na vida é possível com esforço e dedicação. Um especial obrigado à Isabel por toda a paciência e carinho, e finalmente um agradecimento especial ao José Manuel e à Henriqueta pelos conselhos e conhecimento que me transmitiram.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

With the increasing availability of textual information throughout the Web, more and more opportunities are being presented to the areas of Information Extraction and Information Retrieval. Two particularly interesting application areas are opinion mining and geographical text mining. My thesis relates to exploring automated techniques to identify the geographical location that best describes the content of textual documents, with the objective of building a system that discovers and maps opinions towards certain themes, expressed in the context of particular locations, with basis on information extracted from textual documents. This system has three major modules, namely one for georeferencing textual documents, another for mining opinions expressed in textual documents, and finally a module for the construction of maps with the geographical distribution of opinions. In my MSc thesis, I studied and compared techniques to address the challenges associated with all these three modules, although the most important contributions are in the context of the first module (i.e., on document georeferencing).

Most textual documents can be said to be related to some form of geographic context and, recently, Geographical Information Retrieval (GIR) has captured the attention of many different researchers that work in fields related to retrieving and mining contents from large document collections. We have, for instance, that the task of resolving individual place references in textual documents has been addressed in several previous works, with the aim of supporting subsequent GIR processing tasks, such as document retrieval or cartographic visualization of textual documents (Lieberman & Samet, 2011). However, place reference resolution presents several non-trivial challenges (Amitay *et al.*, 2004; Leidner, 2007; Martins *et al.*, 2010), due to the inherent ambiguity of natural language discourse (e.g., place names often have other non geographic meanings, different places are often referred to by the same name, and the same places are often referred to by different names). Moreover, we have that there are many vocabulary terms,

besides place names, that can frequently appear in documents related to specific geographic areas, and GIR applications can also benefit from this information. Instead of trying to correctly resolve the individual references to places that are made in textual documents, it may be interesting to study instead methods for assigning entire documents to geospatial locations (Adams & Janowicz, 2012; Wing & Baldridge, 2011). In the context of my thesis, I have studied methods, based on language model classifiers, for georeferencing entire documents with basis on the raw textual contents.

The extraction of opinions from textual documents is another increasingly important area of research. There are many applications for this new technology, such as the classification of movie and book reviews for the purpose of building automatic recommendations. With the increasing number of opinionated documents available on the Internet, it becomes increasingly important to analyze the underlying opinions expressed in their contents. Opinion mining also presents many challenges and new problems, which throughout this work I also addressed and tried to solve. Opinion mining can be, for instance, done at many levels, namely at the document level, sentence level or phrase level. This work essentially addressed the classification of opinions at a document level also, through the usage of language model classifiers.

In terms of maps, they are the primary mechanism for summarizing and communicating geographically related information. There are many types of thematic maps, used to effectively represent different types of information. Using the geographical information extracted from textual documents, and using the opinions expressed in the same documents, I constructed thematic maps that represent the geographical distribution of opinions, plotting the density of particular opinion classes over the study regions.

## 1.1 Hypothesis and Methodology

In the context of my MSc thesis, I propose to test the hypothesis that *through Information Extraction and Information Retrieval techniques it is possible to find the geographical distribution of opinions, from a given collection of textual documents, and we can later represent this distribution in a thematic map*. In order to test this hypothesis, I constructed a prototype system composed of three modules namely, a document georeferencing module, an opinion classification module, and a map construction module.

In order to test my prototype system and validate my hypothesis, I made two types of tests, namely tests with the first two modules, in order to access their performance under different configurations, and a complete prototype test.

## 1.2 Contributions

The research made in the context of my MSc thesis led to the following main contributions:

- I studied techniques for georeferencing textual documents, based only on the raw text as evidence. The studied techniques relied on a hierarchical classification approach based on either token-based or character-based language models, and on a discretization of the Earth's surface based on the Hierarchical Triangular Mesh approach. Four different post-processing methods were explored in order to assign coordinates to textual documents, with the results showing that the post-processing method that uses the coordinates from the 5 most similar training documents, presents better results, with an average distance of 265 Kilometers, and a median distance error of 22 Kilometers.
- I studied techniques for mining opinions expressed in textual documents, either considering a two-point opinion scale or a five-point scale. The studied techniques relied on either token-based or character-based language models. In the five-point scale case, a hierarchical classification approach was explored, in order to increase the computational performance. Also, a meta-algorithm that corrects the initial labeling of the classifier, known as metric labeling, was considered for the case of the five-point opinion scale. The best performing method for both scales uses character-based language models. In the two-point scale case achieved an accuracy of 0.80, while the best performing method for the five-point scale case achieved an accuracy of 0.5 using the hierarchical classification approach.
- I studied techniques based on kernel density estimation, in order to create thematic density maps that represent the geographical distribution of opinions, where the results show that the maps obtained through automatic extraction correspond to an accurate representation for the geographic distribution of opinions.

In order to make available the research done in the area of geographical information retrieval, the proposed method for document georeferencing as been published in the Spanish Conference of Information Retrieval (Dias *et al.*, 2012). More recently, I also submitted a second article, about document georeferencing which includes three post-processing methods to assign coordinates to textual documents, into the Portuguese journal for the automatic processing of the Iberic languages (Linguamática). The document georeferencing module source code has been shared in Google Code[1], in order to make it available to other researchers that work in the same field, a demonstrator for the document geocoding module has also been made available online[2].

[1] `http://code.google.com/p/document-geocoder/`
[2] `https://appengine.google.com/dashboard/nonedeployed?app_id=s~lm-geocoder`

## 1.3 Document Organization

The rest of this dissertation is organized as follows: Chapter 2 presents the theoretical foundations and the most important related work, including work on geographic information retrieval and on opinion based document classification. Chapter 3 details the proposed methods and the main contributions, separately describing the techniques for georeferencing textual documents, the opinion mining approaches, and the methods for generating thematic maps from the extracted information. Chapter 4 presents the experimental validation of the thesis proposal. Finally, Chapter 5 presents the main conclusions and discusses possible directions for future work.

# Chapter 2

# Concepts and Related Work

This chapter describes the fundamental concepts necessary to understand the research work that has been performed in the context of my MSc thesis. It also presents the most important related work, focusing on the text mining problems of document georeferencing and opinion-based document classification.

## 2.1 Fundamental Concepts

As mentioned in the first chapter of this dissertation, my MSc thesis relates to the area of Information Extraction, which concerns with the extraction of structured information from textual documents. There are several subtasks within this general area. This work focuses explicitly on the task of document classification, namely the classification of documents according to their geographic location, and the classification of documents according to the overall opinion expressed. Moreover, we have that before using advanced techniques for extracting information from documents, one usually pre-filters the interesting documents through Information Retrieval approaches.

The following section introduces important concepts from the areas of Information Extraction and Retrieval. Section 2.1.2 describes two popular models for addressing the task of Named Entity Recognition in textual documents, while Section 2.1.3 concerns with important concepts and approaches for adressing the task of classifying documents into classes. Section 2.1.4 describes important topics from thematic cartography. Finally, Section 2.1.5 describes the most popular evaluation metrics used in the areas of Information Extraction and Information Retrieval.

### 2.1.1 Keyword Based Document Retrieval

There are several techniques that one can use to retrieve relevant documents based on keywords, from a large document collection. The main idea of these techniques is to use a common representation for the textual documents and the query keywords. Using this common representation, the task of retrieving relevant documents resumes to finding the most similar document to the query. One of the most popular methods used for representing textual documents is based on transforming documents into vectors of terms (a.k.a tokens), where terms are the fundamental units from the text (e.g., words or $n$-grams). Each term appears only once in the vector and has a different weight, depending on, for instance, the number of times the term appears in the document (i.e., if terms are defined to be words, then the vector will have a length equal to the number of distinct words in the vocabulary used in the document collection). There are several methods for defining the term weights, including the Term Frequency–Inverse Document Frequency (TF-IDF) method (Manning *et al.*, 2008).

In brief, we have that the TF-IDF method calculates the weight of each term in a term vector for a given document. The weight of each term indicates the importance of the term for a document, within a set of documents. If a term appears many times within a document, than this term is highly descriptive for the document's contents. However, if this term also appears often in many other documents, this may indicate that the term is not so important. Thus, the TF-IDF method gives more importance to terms that are very frequent in a certain document, but that are rare in the rest of the documents contained in the corpus.

Formally, we have that the term frequency in a certain document is given by:

$$tf_{t,d} = \frac{n_{t,d}}{k_d} \tag{2.1}$$

In the formula, $n_{t,d}$ represents the number of terms $t$ found in the document $d$, and $k_d$ represents the total number of terms in document $d$.

The inverse document frequency is computed by taking the logarithm of the division between the total number of documents contained in the corpus, and the number of documents that contain the term, represented in Equation 2.1 by $df_t$.

$$idf_t = log\left(\frac{|D|}{1 + df_t}\right) \tag{2.2}$$

Thus, each term's weight is given by :

$$tf\_idf_{t,d} = \frac{n_{t,d}}{k_d} \times log\left(\frac{|D|}{1+df_t}\right) \tag{2.3}$$

Now that we have the weight of each term in a document, we can view a document as a vector of terms with a term weight given by Equation (2.3). To search for relevant documents, we can start by representing a query $q$ as a vector of terms, through the same representation that we used for documents. To query a collection of documents, we can compute the similarity between the query and each document, using for instance the cosine similarity metric, which is given by:

$$score(q,d) = \frac{\overrightarrow{q}.\overrightarrow{d}}{\|\overrightarrow{q}\|\,\|\overrightarrow{d}\|} = \frac{\sum_{i=1}^{n}(q_i \times d_i)}{\sqrt{\sum_{i=1}^{n}(q_i)^2} \times \sqrt{\sum_{i=1}^{n}(d_i)^2}} \tag{2.4}$$

In the formula, the nominator is the dot product between the document vector and the query vector, while the denominator is the product of the Euclidean lengths of the vectors. The denominator normalizes the dot product of the vectors in order to compensate for their length. This normalization is necessary because, while two vectors can encode a similar relative term distribution, the document vector is usually much larger than the query vector. Using this technique we measure the similarity between each document in the collection and the query, usually returning the $K$ most similar documents to the query.

Although this method is very precise, it can be computationally very expensive. Computing a single similarity between a document and a query can entail a dot product in thousands of dimensions, demanding thousands of arithmetic operations. A faster method is achieved by using a sum of the TF-IDF weights of a document's terms, for each term present in the given query:

$$score'(q,d) = \sum_{t \epsilon q} tf\_idf_{t,d} = \sum_{t \epsilon q}\left(\frac{n_{t,d}}{k_d} \times log\left(\frac{|D|}{1+df_t}\right)\right) \tag{2.5}$$

### 2.1.2 Named Entity Recognition in Text

Named Entity Recognition (NER) is a particular task of Information Extraction. Its purpose is to classify the words in a text into certain categories, such as names of persons, locations or organizations. Although current approaches for NER can achieve a near human accuracy, several problems still exist. It is important to notice that the task is particularly challenging, due to the high ambiguity of some of the entities that occur many times in a document collection, and which can have many classifications (e.g., *Washington* the state, the city or the person). A particular challenge related to the objective of this work is to identify locations. There are several algorithms

to recognize entities in texts. Examples include dictionary based algorithms, algorithms based on rules, and algorithms based on Machine Learning (Krovetz *et al.*, 2011). This section presents two specific models that are widely used when addressing the NER task through Machine Learning, namely Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs).

#### 2.1.2.1 Hidden Markov Models in NER Problems

A Hidden Markov Model is a statistical Markov model for sequences of observations (e.g., text tokens), which considers hidden states. In this model, only the tokens are visible. The conditions that influence the generation of tokens are hidden, which means that we do not know directly the sequence of states being modeled. To better understand this model consider a typical example of an information extraction task, where we would want to analyze and extract place references from the following sentence from a news article: *Congo gained independence from Belgium in 1960*. HMMs can be seen as state machines, where the states represent the types of fields we want to extract, and each state can generate a certain number of tokens, as seen in Figure 1. In this example, each token can be classified as a non-target token or a target token (i.e., non place references and place references). The parameters of the model are the probabilities of starting at a particular state, the transition probabilities from a state to another, and the probabilities of a state generating a certain token. Given a sequence of observations (i.e., the tokens from a textual document), we can classify each token by determining the most likely sequence of states that could have generated the sequence of tokens.

Considering the previous example, we can use a notation where $\lambda = (A, B, \pi)$ is the HMM, $\pi_i$ is the probability of being in the state $i$ at the beginning of the experience, $A_{ij}$ is a matrix with the probabilities of transiting from a state $i$ to a state $j$, and where $B_{jk}$ is a matrix with the probabilities of observing a certain token $k$ when in a state $j$. Also consider that $N$ is the number of states in the model (which in the example corresponds to place references and non place references), $M$ is the number of distinct tokens, $T$ is the length of the observed sequence (which in the example is number of words in the sentence), $i_t$ represents the state that we are in at a time $t$, $O = o_1, o_2, ..., o_T$ is the sequence of observed tokens, $\Sigma$ is the set of distinct observed tokens, and $Q$ is the set of possible transitions between states.

In the context of NER, the two most important problems to solve with HMMs are (i) computing the most likely sequence of states that originated a certain sequence of observed tokens, and (ii) training the HMM using a series of observations and/or state sequences.

Formally, the first problem can be described as, given the HMM $\lambda = (A, B, \pi)$, choosing the state sequence $I = i_1, i_2, ..., i_T$ that maximizes $P(O, I|\lambda)$ given a sequence of observed tokens $O$.

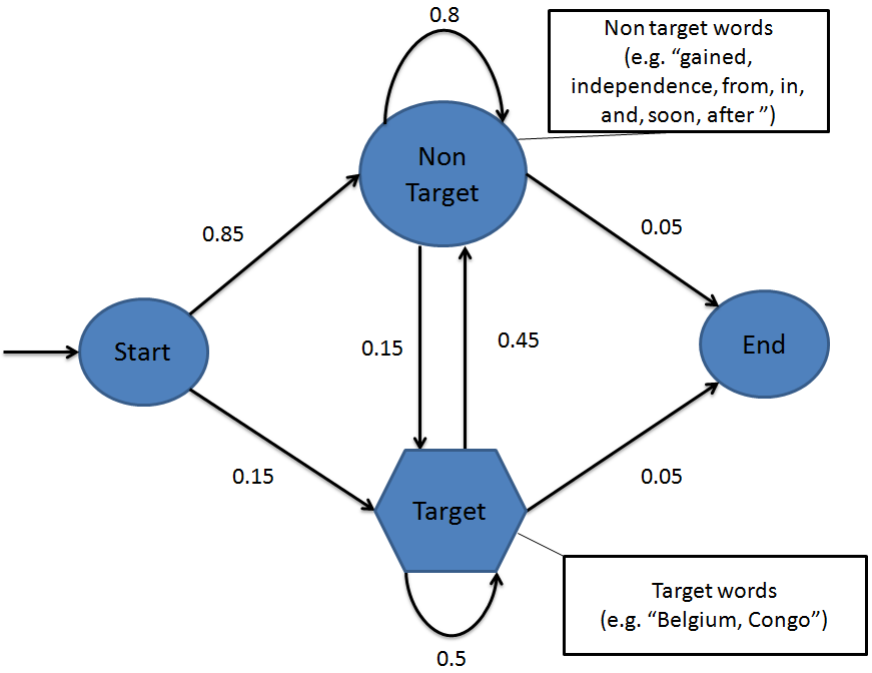



**Figure 2.1:** An HMM seen as a state machine.

This problem can be solved using the Viterbi Algorithm (Viterbi, 2006). In brief, we have that the Viterbi algorithm is a dynamic programming approach for computing the least costly path over the possible state sequences for generating the observed sequence. The total cost of the path is the sum of the weights of all the edges we cross. Note that the joint probability of observing a certain sequence of tokens and a certain sequence of states is given by:

$$P(O, I|\lambda) = P(O|I, \lambda).P(I|\lambda) = \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2)...a_{i_{t-1} i_t} b_{i_t}(o_T) \tag{2.6}$$

To simplify the computation we can transform the multiplications into sums of logarithms, thus avoiding numerical precision issues:

$$U(i_1, i_2, ..., i_T) = -[ln(\pi_{i_1} b_{i_1}(o_1)) + ln(a_{i_1 i_2} b_{i_2}(o_2))...ln(a_{i_{t-1} i_t} b_{i_t}(o_T))] \tag{2.7}$$

Then, we can say that the cost of transiting from a state $i$ to a state $j$, at time t, is given by the quantity $-ln(a_{i_j i_k} b_{i_k}(o_t))$. For each state, the algorithm computes the cost of transiting from that state in time $t-1$ to every state. The algorithm saves the minimal cost to reach every state and the origin state. This procedure is repeated $T$ times. In the end, the algorithm chooses the path with the least cost, thus resulting in the corresponding sequence of states.

The second problem, training the HMM, is usually addressed by using training data to estimate the transition and emission probabilities. The training data is a set of previously labeled documents, where each document has each word labeled. To estimate the transition probabilities, and for each transition in the set of transitions $Q$ from the training set, we can compute:

$$P(q \to q') = \frac{c(q \to q')}{\sum_{s \epsilon Q} c(q \to s)} \tag{2.8}$$

In the formula, $c(q \to q')$ is the number of transitions from state a $q$ to a state $q'$ in the document set, and $c(q \to s)$ is the number of transitions from state $q$ to any state. To estimate the emission probabilities, for each token in the token set $\Sigma$ that exists in the training data, we can compute:

$$P(q \uparrow \sigma) = \frac{c(q \uparrow \sigma)}{\sum_{p \epsilon \Sigma} c(q \to p)} \tag{2.9}$$

In the formula, $c(q \uparrow \sigma)$ is the number of times that a token $\sigma$ was generated by a state $q$ and $c(q \to p)$ is the total number of tokens generated in state $q$.

For more information about HMMs please refer to Rabiner (1989) and Dugad & Desai (1996).

#### 2.1.2.2 Conditional Random Fields in NER Problems

Although HMMs are a natural choice to model problems such as named entity recognition, recently it has become common to address the task through more powerful sequence modeling approaches, an example being Conditional Random Fields (CRF). A CRF is essentially a discriminative undirected graphical model in which each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. In the case of sequence labeling problems such as named entity recognition, linear-chain CRFs are typically used, in which an input sequence of observed variables $X$ represents a sequence of observations (i.e., word tokens) and $Y$ represents a sequence of hidden states (i.e., the labels for each token) that need to be inferred given the observations. The $Y_i$s are structured to form a chain, with an edge between each $Y_{i-1}$ and $Y_i$. The underlying idea is that of defining a conditional probability distribution over label sequences $Y$, given a particular sequence of word tokens, rather than a joint distribution over both label and observation sequences as in the case of HMMs. The conditional dependency of each $Y_i$ on $X$ is defined through a fixed set of feature functions of the form $f(i, Y_{i-1}, Y_i, X)$, which can informally be thought of as measurements on the input sequence that partially determine the likelihood of each possible value for $Y_i$. The model assigns each feature a numerical weight, and combines them to determine the probability

of a certain value for $Y_i$. The conditional probability distribution in first-order linear chain CRFs is given by the following equation:

$$p(Y|X,\lambda) = \frac{1}{Z(X)} exp\left(\sum_j \lambda_j \sum_i^{|X|} f_j(i, Y_{i-1}, Y_i, X)\right) \tag{2.10}$$

In the formula, $Z(X)$ is a normalization factor and $\lambda$ is a parameter to be estimated from the training data. Notice that CRFs have many similarities towards the conceptually simpler Hidden Markov Models (HMMs), but relax certain assumptions about the input and output sequence distributions. An HMM can loosely be understood as a CRF with very specific feature functions that use constant probabilities to model state transitions and emissions. Conversely, a CRF can loosely be understood as a generalization of an HMM that makes the constant transition probabilities into arbitrary functions that vary across the positions in the sequence of hidden states, depending on the input sequence. Notably in contrast to HMMs, we have that (i) CRFs can contain any number of feature functions, (ii) the feature functions can inspect the entire input sequence at any point during inference, and (iii) the range of the feature functions need not have a probabilistic interpretation. Linear-chain CRFs can be used through essentially the same techniques that were presented in the case of HMMs. To compute the most probable assignment for $Y$, given a certain input $X$, we can use the Viterbi algorithm, where the constant probabilities of HMM transitions correspond to the feature functions of the CRF.

The training of the CRF models can be done through many techniques, an example being Stochastic Gradient Descent (SGD) (Wijnhoven & de With, 2010). SGD is a simple optimization method that is designed to exploit the assumption that many items from the training data provide similar information about the parameters of the model. Using this assumption, it is possible to update the parameters after seeing only a set of few examples, instead of sweeping through all of them.

For more information about CRFs please refer to the article by Sutton & Mccallum (2006) and to the article by Vishwanathan *et al.* (2006).

### 2.1.3 Document Classification

Classifying documents according to the polarity of the expressed opinions, or classifying documents according to their geographical location, can be seen as particular cases of traditional document classification tasks. This section describes two widely used approaches for document classification, namely Support Vector Machines, and classifiers based on Language Models.

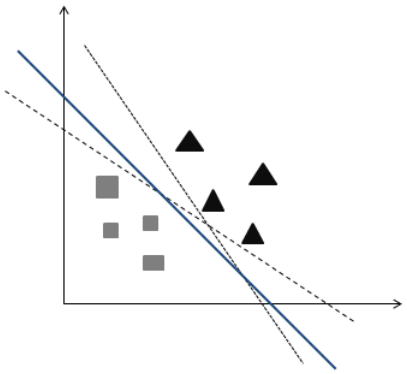


**Figure 2.2:** Example of how two types of data can be separated by many different hyperplanes.

#### 2.1.3.1 Support Vector Machines

Support Vector Machines (SVMs) are a technique for supervised classification created by Vapnik (1979). In the conventional model, the SVM classifier assigns a given input into one of two classes. The basic idea behind this algorithm is to construct a hyperplane that separates the input, so that the margin of separation of the data is maximum (Boser *et al.*, 1992).

As shown in Figure 2.2, there may be multiple hyperplanes that separate two classes in a dataset (i.e., triangles and rectangles), but only one that separates the data with the maximum margin of separation between the two classes. We want to separate each data point $x_i$ in the dataset $x$, and classify these points into one of the two classes, $y_i \epsilon \{-1, +1\}$, where $-1$ and $+1$ represent the two types of data. We define $w$ as a normal hyperplane vector which is perpendicular to the hyperplane separating the classes. This vector is know in the SVM literature as the weight vector. We also define $b$ as an interceptor term which defines the separating hyperplane. Because the normal vector is perpendicular to the hyperplane, all points in the vector $x$ satisfy $w^T x = -b$. Using the $sign$ function, for every point in each class we have that $|sign(w^T x + b)| = 1$ . We also have that each point $x_i$ belongs to one of the two classes according to:

$$sign(w^T x_i + b) \begin{cases} = +1 & \text{, then } y_i = +1 \\ = -1 & \text{, then } y_i = -1 \end{cases} \tag{2.11}$$

Points $x_i$ that satisfy $w^T x + b = 1$ are called the support vectors, represented in Figure 2.3 by the red squares and circles. The support vectors are the only data points that determine the hyperplane, and hence the name support vectors. This property enables the SVM classifier to achieve good results in ill-posed problems (e.g., when dealing with data sets with a low ratio of

sample size to dimensionality), since the separation margin only depends on the support vectors. The distance between each support vector and the hyperplane $w$ is equal to $1/||w||$. In order to maximize this distance, we can to solve the following quadratic optimization problem:

$$\begin{array}{ll} \min\limits_{w,b} & ||w||^2 \\ \text{subject to (s.t.)} & y_i(w^T x + b) \geq 1 \text{ for each } i = 1, \ldots, n \end{array} \tag{2.12}$$

To maximize the margin we use $||w||^2$, so as to have a quadratic minimization problem. Quadratic optimization problems are a well known class of mathematical optimization problems. The solution involves the construction of a dual optimization problem, where a Lagrange multiplier $\alpha_i$ is associated with each constraint $y_i(w^T x + b) \geq 1$. The problem can be translated into finding $\alpha_1 ... \alpha_N$ such that $\sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j$ is maximized, and so that:

$$\sum_i \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0 \text{ for all } 1 \leq i \leq N \tag{2.13}$$

The solution corresponds to:

$$\begin{array}{c} w = \sum_i \alpha_i y_i x_i \\ \text{and} \\ b = y_k - w^T x_k \text{ for any } x_k \text{ such that } \alpha_k \neq 0 \end{array} \tag{2.14}$$

Each $\alpha_i \neq 0$ indicates that the corresponding $x_i$ is a support vector. The classification function is finally given by the following equation:

$$f(x) = sign\left(\sum_i \alpha_i y_i x_i^T x + b\right) \tag{2.15}$$

Notice that SVMs are linear classifiers. In order to solve nonlinear problems, Vapnik *et al.* proposed an extension by applying the Kernel trick (Boser *et al.*, 1992). In Figure 3, we have the initial space of an example problem, which is non-linearly solvable. The idea is to map the space into another feature space, usually of higher dimensionality, so we can separate the data linearly.

Suppose we want to map the space of the problem into a new feature space, according to a transformation $\phi : x \mapsto \phi(x)$. This mapping can be done using a Kernel function, which acts as a dot product in a feature space. More formally, a Kernel function $K(x, y)$ is a function such that $K : X \times X \mapsto R$, and for which there exists a function $\phi : X \mapsto Z$, where $Z$ is a real vector space such that $K(x, y) = \phi(x_i)^T \phi(x_j)$. For example, consider two dimensional vectors $a = (a_1, a_2)$ and $b = (b_1, b_2)$, and consider the Kernel function $K(a, b) = (1 + ab)^2$. We would then have that :

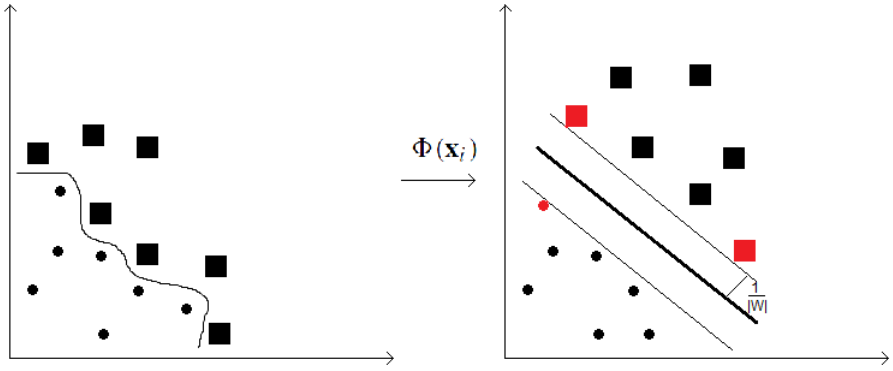



**Figure 2.3:** Transformation from the problem space into a feature space

$$\begin{aligned} K(a,b) = (1+ab)^2 &= 1 + a_1^2 b_1^2 + 2a_1 b_1 a_2 b_2 + a_2^2 b_2^2 + 2a_1 b_1 + 2a_2 b_2 \\ &= (1, a_1^2, \sqrt{(2)} a_1 a_2, a_2^2, \sqrt{(2)} a_1, \sqrt{(2)} a_2)^T \\ &\quad (1, b_1^2, \sqrt{(2)} b_1 b_2, b_2^2, \sqrt{(2)} b_1, \sqrt{(2)} b_2) \\ &= \phi(a)^T \phi(b) \end{aligned} \tag{2.16}$$

The data point $\phi(a)$ will be transformed into a higher-dimensional vector which corresponds to $(1, a_1^2, \sqrt{(2)} a_1 a_2, a_2^2, \sqrt{(2)} a_1, \sqrt{(2)} a_2)$, and the transformed data point $\phi(b)$ is the high-dimensional vector $(1, b_1^2, \sqrt{(2)} b_1 b_2, b_2^2, \sqrt{(2)} b_1, \sqrt{(2)} b_2)$. In this case, $\phi : R^2 \mapsto R^6$. We can see from this example that the mapping of a data points $x$ to $\phi(x)$ can be very costly (e.g., in the case of radial basis functions, the feature space has infinite dimensionality). However, notice that the dot product present in our classification problem $f(x)$ becomes $\phi(x_i)^T \phi(x_j)$. We can compute the value of this dot product directly, instead of mapping $x \mapsto \phi(x)$ using the Kernel function, and so our classification function in the feature space becomes $f(x) = sign(\sum_i \alpha_i y_i K(x_i, x) + b)$. In SVMs, one of the most used Kernel function families are the radial basis functions, particularly the Gaussian Radial Basis Function corresponding to:

$$K(x,y) = e^{-(x-z)^2/2\sigma} \tag{2.17}$$

For more detailed information about SVMs please refer to the book by Manning *et al.* (2008), Moguerza & Muntildeoz (2006), or to the paper by Boser *et al.* (1992).

#### 2.1.3.2 Language Model Classifiers

Language models are a natural choice to model text categorization problems. For each category, we can build a language model that, for each unseen sequence of words or characters (*i.e.*, for each document), returns the probability of that sequence having been generated by the corresponding language model (*i.e.*, by a particular class). Language models have many advantages in comparison to other types of classifiers. They are both simple and fast and, unlike SVMs, they require no feature engineering and almost no pre-processing of data. Thus, unlike in the case of classifiers based on features, language model classifiers can be applied to many languages without any extra effort (Peng *et al.*, 2003).

In brief, we have that a language model is a probabilistic function that assigns probabilities to strings of symbols. These strings can be either a sequence of characters or a sequence of tokens. Usually in classification problems, language models are based on *n*-grams (*i.e.*, a sequence of $n$ symbols), although in the area of Information Retrieval it is common to use only unigrams (Manning *et al.*, 2008). The basic idea in a $n$-gram based language model is to build a vocabulary of $n$-grams and to count the number of different $n$-grams present in a textual document or collection of documents. These counts are used to infer a multinomial distribution for the $n$-grams, and later used to infer the probability of a new sequence of symbols. $N$-gram models capture the probability of a symbol $s$ based only on the previous $n-1$ symbols.

$$P(s_1, ..., s_m) = \prod_{i=1}^{m} P(s_i | s_{i-n+1}, ..., s_{i-1}) \tag{2.18}$$

In order to deal with the possibility of under-flows from multiplying long sequences of numbers that are less than 1, most implementations of language models use the log of probabilities.

$$log(P(s_1, ..., s_n)) = \sum_{i=1}^{n} log(P(s_i | s_{i-n+1}, ..., s_{i-1})) \tag{2.19}$$

An estimate of $n$-gram probabilities for a given corpus is given by the frequency of each of the patterns, according to:

$$P(s_i | s_{i-n+1} ... s_{i-1}) = \frac{\#(s_{i-n+1} ... s_i)}{\#(s_{i-n+1} ... s_{i-1})} \tag{2.20}$$

Notice that it is likely that, for unseen documents, we will encounter unseen $n$-grams, which therefore will have a probability of zero. A smoothing technique is necessary for assigning non-zero probabilities to these $n$-grams. One standard approach is to use some sort of back-off estimator.

For a given unseen $n$-gram, we compute the probability of a subset of that $n$-gram multiplied by a normalization constant $\beta(s_{i-n+1}...s_{i-1})P(s_i|s_{i-n+2}...s_{i-1})$. For the non-zero probability $n$-grams, we can compute a discounted probability using a smoothing technique like absolute smoothing, Good-Turing smoothing, linear smoothing, or Witten-Bell smoothing (Chen & Goodman, 1996).

In order to use language models for classification, we compute a language model for each category $c \epsilon C = \{c_1, ..., c_t\}$, using a collection of documents to train the model. For an unseen document, we compute the probability of that document having been generated by each language model. The language model that returns the highest probability is the best category.

$$arg\ max_{c \epsilon C} \left\{ \sum_{i=1}^{n} log(P_c(s_i|s_{i-n+1}, ..., s_{i-1})) \right\} \tag{2.21}$$

Some authors have proposed extensions to the classification method explained previously, using the joint probability between the probability for each language model and a multinomial probability distribution over all categories, so that the most frequent categories in the data are also more likely to be assigned to the documents (Carpenter & Baldwin, 2011).

### 2.1.4 Mapping Geographical Phenomena

Maps are the primary mechanism for summarizing and communicating geographically related information. In the context of my MSc thesis, I developed a system capable of generating maps that show the geographic distribution of opinions towards certain topics, as extracted from textual documents. This section summarizes the most common thematic mapping techniques.

In brief, we have that cartographers commonly distinguish between point, area and line symbolizations (Slocum *et al.*, 2005). Different thematic mapping techniques (i.e., techniques for producing maps or charts that show particular themes connected to specific geospatial areas) can use these symbols to effectively map geographical phenomena, in a way that is easily perceived.

Point symbols refer to particular locations in geographic space, and they are commonly used when the geographical phenomena being mapped is located at a specific place or is aggregated to a given location. Differentiation among point symbols is achieved by using visual variables such as size, color, transparency and shape. Common thematic mapping techniques using point symbols are dot maps and proportional symbol maps. On a dot map, one dot represents a unit of some phenomena, and dots are placed at locations where the phenomenon is likely to occur. A proportional symbol map is constructed by scaling the symbols in proportion to the magnitude of the values occurring at particular point locations. These locations can be true points or conceptual points, such as the center of a country for which the data have been collected.

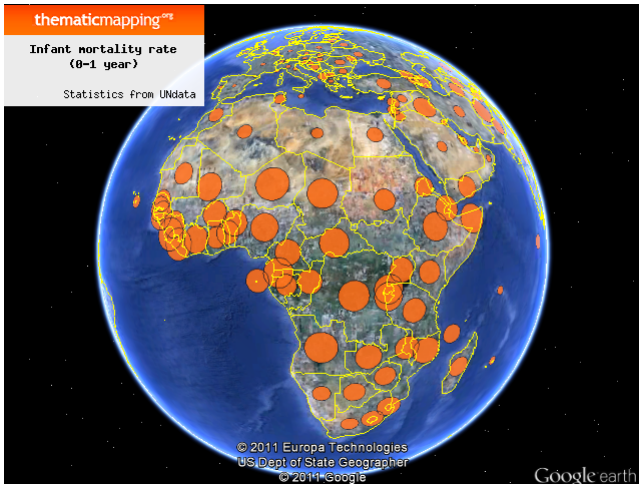



**Figure 2.4:** Example of a proportional symbols map.

Figure 4 presents an example of a proportional dot map, built with Google Earth and with the thematic mapping engine[1] developed by Sandvik (2008) in the context of his MSc thesis.

Area symbols are used to assign a characteristic or value to a whole area on a thematic map. They can be used to better depict a variable which cannot be measured at a point, but which instead must be calculated from data collected over an area. An example would be population density, which can be calculated by dividing the population of a statistical reporting unit by the surface area of that unit. Visual variables used for area symbols are color, texture and perspective height. The choropleth map is probably the most commonly employed method of thematic mapping, and is used to portray data collected for enumeration units specified as enclosed regions, such as countries or other political/statistical reporting units. While choropleth maps reflect the structure of the data collection units, isarithmic maps (also known as contour maps, density maps, heat maps or isopleth maps) depict smooth continuous phenomena, such as elevation or barometric pressure. On this type of maps, line-bounded areas can for instance be used to represent regions with the same value (e.g., on an elevation map, each elevation line indicates an area at the listed elevation). Dasymetric maps offer a compromise between choropleth maps and isarithmic maps, assuming areas of relative homogeneity separated by zones of abrupt change. A dasymetric map is similar to a choropleth map, but the regions are not predefined and instead chosen so that the distribution of the measured phenomenon, within each region, is relatively uniform. Figure 5 presents an example of a choropleth map.

It is important to notice that area symbols, particularly on isarithmic maps, can also be used

[1] http://thematicmapping.org/engine

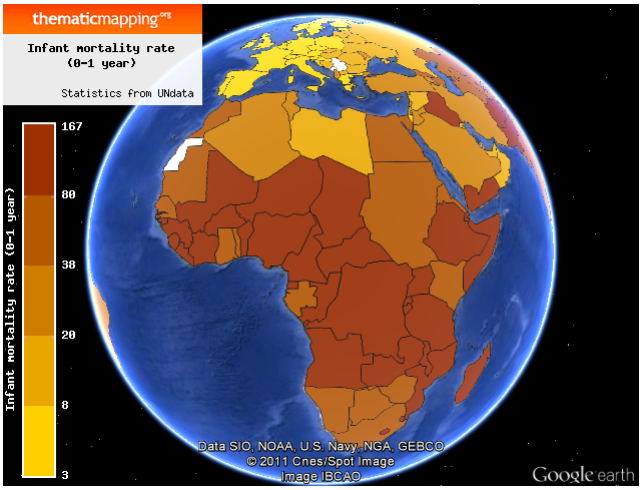



**Figure 2.5:** Example of a choropleth map.

when we have a very large number of data points, making it difficult to identify coherent patterns. Continuous surfaces can be produced from point data, through spatial interpolators such as inverse-distance weighting or Kriging (i.e., in the case of sampled data (Li & Heap, 2008)), or through density estimators which spread out the data from each point over the surrounding area (e.g., through the Kernel density estimation technique (Carlos *et al.*, 2010)).

Finally, we have that line symbols are used in thematic maps to indicate connectivity or flow, to indicate equal values along a line, and to show boundaries between unlike areas. Line symbols are differentiated on the basis of their form (e.g., solid lines vs dotted lines), color and width. Common thematic mapping techniques that use line symbols include flow maps, which use lines of differing width to depict the movement of phenomena between geographic locations. Isarithmic maps are also often categorized as maps using line symbolizations, as they can be based on line-bounded areas to depict continuous phenomena.

In the context of his MSc thesis, Bjørn Sandvik studied how these techniques can be used together with the Keyhole Markup Language (KML), *i.e.* the XML format used in the popular Google Earth virtual globe (Sandvik, 2008). In the context of my MSc thesis, I used the mapping functionalities available on the R[1] project for statistical computing, essentially mapping continuous surfaces obtained through Kernel density estimation.

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

### 2.1.5 Evaluation Metrics for Information Extraction and Retrieval

When evaluating the effectiveness of document retrieval or classification methods, the most frequently used measures are precision and recall. In order to simplify the presentation, consider the retrieval case. Precision measures the fraction documents that were correctly retrieved against the total number of retrieved documents. More formally, precision is given by:

$$Precision = \frac{\#relevant\ items\ retrieved}{\#retrieved\ items} \tag{2.22}$$

Recall measures the fraction of relevant documents that were retrieved against the total number of relevant items in the collection of documents. More formally, recall is given by:

$$Recall = \frac{\#relevant\ items\ retrieved}{\#relevant\ items} \tag{2.23}$$

Although these measures are important to evaluate a system, they alone are not sufficient. The documents that a system retrieves can all be relevant, although a large portion of relevant documents is left unretrieved. In this case the system will have a high precision but a low recall. A system may also retrieve all relevant documents, but with many of the retrieved documents not being relevant. In this case the system will have a high recall but low precision. In order to evaluate a system effectively, we need a measure that relates precision and recall. The commonly used F-measure offers just that, corresponding to the weighted harmonic mean of precision and recall. The F-measure corresponds to the following equation:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \text{ where } \beta^2 = \frac{1-\alpha}{\alpha} \tag{2.24}$$

The default value for $\beta$ is $1$, equally weighting precision and recall. This metric is commonly referred to as $F_1$, and it can be simplified into:

$$F_1 = \frac{2PR}{P+R} \tag{2.25}$$

Using values for $\beta < 1$ emphasizes precision, while using values for $\beta > 1$ emphasizes recall.

In classification problems, we may have to classify items into multiple classes. In the case that the system classifies an item as belonging to a certain class, classification can either be correct (i.e., a true positive) or incorrect (i.e., a false positive). If the system considers that an item

does not belong to a certain class, this can either be correct (i.e., true negative) or incorrect (i.e., false negative). In multi-class problems, to measure the aggregate precision and recall, people frequently use micro-averages and macro-averages. Micro-averaged-precision is given by Equation (2.26), where $|C|$ represents the number of classes :

$$P_{micro} = \frac{\sum_{i=0}^{|C|} TP_i}{\sum_{i=0}^{|C|} TP_i + FP_i} \tag{2.26}$$

In the formula, $TP_i$ represents the true positives, for a class $i$, while $FN_i$ represents the false negatives. Similarly, micro-averaged-recall is given by:

$$R_{micro} = \frac{\sum_{i=0}^{|C|} TP_i}{\sum_{i=0}^{|C|} TP_i + FN_i} \tag{2.27}$$

In the formula, $FP_i$ represents the false positives for a class $i$.

Macro-average metrics are instead based on computing a simple average over all classes. Macro-averaged-precision is given by:

$$P_{macro} = \frac{1}{|C|} \sum_{i=0}^{|C|} \frac{TP_i}{TP_i + FP_i} \tag{2.28}$$

As for macro-averaged-recall, it is given by:

$$R_{macro} = \frac{1}{|C|} \sum_{i=0}^{|C|} \frac{TP_i}{TP_i + FN_i} \tag{2.29}$$

Notice that the difference between the two approaches lies in the fact that, while macro-averages give an equal weight to each class, micro-averages give an equal weight to each item. In micro-averages, small classes can be dominated by larger classes, and the final result will tend to be closer to that of the larger classes. Macro-averages are better suited to measure precision and recall across multiple classes, when the classes have very skewed sizes.

In classification problems, another commonly used measure is accuracy, which is the fraction of correctly classified documents:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.30}$$

Accuracy is not used in information retrieval problems, since most of the times the majority of the documents will belong to the non-relevant class, making the measure ineffective. For more

information about these and similar evaluation metrics, please refer to Manning *et al.* (2008).

Notice that thematic maps built from estimated data can also be evaluated using metrics from the area of information retrieval, such as accuracy. We can for instance divide a region of study into a grid of sub-regions. For each estimated position, we can verify if the sub-region in the grid that was estimated is the same as the true sub-region.

## 2.2 Related Work

This section presents the most relevant previous works in the context of my MSc thesis. Section 2.2.1 describes works related to the problems of recognizing and disambiguating place references in textual documents, while Section 2.2.2 presents works related to the problem of identifying the location that best summarizes the contents of a textual document. Finally, Section 2.2.3 presents works related to the classification of opinions in texts.

### 2.2.1 Place Reference Resolution

The task of place reference resolution concerns with the identification of place references in textual documents, and also with the discovery of the exact location associated to the recognized references. Thus, this task can be divided into two steps, namely the recognition of the place references and the disambiguation of the references. The first problem is strongly related to Named Entity Recognition, which has been studied extensively by the Natural Language Processing community. The recognition problem presents many challenges related to the ambiguity of natural language, due to the fact that many common words in a natural language can also be place names (e.g., *Of in Turkey* and *To* in *Myanmar*). We also have that many place names have other non-geographical meanings (e.g., *Turkey* the country vs *turkey* the animal, or *Gary* in *Indiana* vs *Gary* the name of a person).

The second sub-task relates to the disambiguation of a place name into its actual location on the surface of the Earth. After identifying a geographic reference in a certain text, we must be sure of what is the place being referenced. This problem also presents several non-trivial challenges. For instance, different places can have the same name (e.g., *Lisboa, Portugal* vs *Lisboa, Colombia*) and the same place can have many different names (e.g., *New York City* vs *Big Apple*).

#### 2.2.1.1 Place Reference Recognition

The task of recognizing place names in text was first approached by using global lexicons or gazetteers with location names. For instance Amitay *et al.* created the Web-a-Where system to resolve place names in Web pages, using an approach based on a large world coverage gazetteer (Amitay *et al.*, 2004).

A more recent approach is to use machine learning techniques. Martins et al. used an HMM to recognize place references in journalistic texts (Martins *et al.*, 2010). The HMM used by the authors is an adaptation of the LingPipe[1] machine learning approach for NER. It first starts by tokenizing the input text, and each token is then assigned to a tag encoding the fact that the token is either part of a place reference or not. The implementation used by the authors uses an encoding that is position sensitive (i.e., the BMEWO+ encoding). This encoding tags each token as being an entity (beginning of the entity (B), middle of the entity (M), end of the entity (E), or single token entity (W)), or as not being an entity (O). The non-entity tokens are also subdivided into four different tags, whether the token precedes an entity, succeeds an entity, is preceded and succeeded by an entity or is not preceded nor succeeded by an entity. The position tags support the creation of transition constraints (e.g., B must precede M) which enable the usage of long-distance information about preceding or following words. This approach achieved state-of-the-art results, with $F_1$ scores of 0.791 for a corpus of English journalistic articles.

Lieberman & Samet (2011) presented a multifaceted approach that mixes many techniques, namely Part-of-Speech (POS) tagging (Schmid, 1994), the usage of entity dictionaries, and Conditional Random Fields (CRF) models (Jenny Rose Finkel & Manning, 2005), to recognize references in journalistic articles retrieved from the NewsStand system (Teitler *et al.*, 2008). The goal of using this mixed technique is to increase the recall measure (i.e., the percentage of entities recognized in the text). Although this method will have low precision rates (i.e., the percentage of rightly identified entities), the authors proposed a subsequent task of resolving place references, which will correctly identify the right place entities. At the recognition stage, the authors attributed more importance to recognizing as many place entities as possible, even though some of them might be wrongly identified. The first step of this technique uses a small gazetteer of well known locations (e.g., continents, countries and top level administrative divisions) to recognize prominent locations in the text. The next step is to use a dictionary of entities that commonly appear on news articles, to recognize varied types of entities (e.g., persons, organizations, etc.). The goal of recognizing other entities besides locations is to help resolving geo/non-geo ambiguities. This step also identifies place entities by using cue words patterns (i.e., keywords that serve to

[1] http://alias-i.com/lingpipe

identify entities, such as *Lake X* or *University of X at Y*). In the next step, a POS tagger is used to identify proper-nouns, since locations are proper-nouns. The authors used the TreeTagger implementation[1], a decision tree-based POS tagger. Although this technique will also identify other entities besides locations, it is consistent with the high recall goal of the authors. The last step of place recognition is to use a CRF to recognize and classify entities in the text, through the implementation from the Stanford NLP NER and IE package[2]. The authors considered only the person, organization and location entities. After all the recognition techniques are applied, the authors use filtering rules to treat the entities identified (e.g., in the CRF technique some of the entities can be fragmented, in that the boundaries were chosen incorrectly).

#### 2.2.1.2 Disambiguation of Place References

Early approaches to address the disambiguation of place names, occurring in textual documents, relied on the help of gazetteers and heuristic rules (e.g., if the name *Paris* appears on a text, after querying the gazetteer, disambiguate the reference to the candidate location with more inhabitants). These rules can be divided into three categories, namely prominence rules, context rules and other rules. In his PhD thesis, Leidner (2007) surveyed the most commonly used heuristics in works related to location disambiguation. The following list of heuristics is based on the details from the survey and on other previous works in the area, such as those of Lieberman *et al.* (2010) and of Martins *et al.* (2010).

Prominence rules are based on the importance of the candidate disambiguations. They favor important places around the world and assume that if some place name occurs, there is a higher probability that the place being referenced is a prominent one. Prominence rules include:

- Population count: If the place reference to be disambiguated has several candidates, the correct place is considered to be the one with the highest number of inhabitants.
- Preference to higher-level references: If a place reference has two candidates, and if one of them is a country and the other one is a city, choose the higher-level unit (i.e., the country).
- Ignore small places: Reduce the size of the database, discarding cities with a number of inhabitants that is less than a certain threshold. This heuristic decreases ambiguity, which increases precision, but it also decreases the recall.
- Frequency weighting: Give more importance to candidates that are more frequent in a document. This heuristic is used when considering decisions about multiple place references,

[1] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger
[2] http://www-nlp.stanford.edu/software/CRF-NER.shtml

and where some of them are non-ambiguous.

- Default referent: Use existing information about the most prominent place reference and disambiguate the place accordingly (e.g., eventhough the state of *New York* has more inhabitants than the city of *New York*, people tend to associate *New York* to the city).
- Number of alternative names: The number of alternative names that a place has is highly linked to its importance. Disambiguation should thus be made to the candidate with more alternative names.

Context rules include the evaluation of clues left in the document by the author, so that the reader can correctly identify the place being mentioned. Based on the assumption that place references in the same text have some kind of relation, these rules try to disambiguate other references. Commonly used context rules include:

- Contained in: When a place name is ambiguous, authors tend to leave clues, so that the readers can identify the place (e.g., clues such as *the event occurred in Paris,Texas* ). If a place name is followed by a *contained in pattern* (e.g., *Lisbon, Portugal* or *London in United Kingdom*) and if there exists a single candidate that satisfies that pattern, then disambiguate the place name with that candidate.
- Superordinate mention: If a place reference $p_1$ is to be resolved, and in the same text there exists a place reference $p_2$ which is already disambiguated to a place that has some relation to a candidate of $p_1$, then assign $p_1$ to this candidate. In an example sentence such as *Last summer I went to Paris, but I never thought that Texas could be so hot*, notice that by knowing that *Texas* refers to the state in the *United States*, we discover that the mention to *Paris* is actually the city of *Paris* in *Texas*. This heuristic is basically a long-distance version of the contained-in heuristic.
- Geometric minimality: Compute the area of the convex hull of all combinations for place name candidates present in the text, and choose the ones that minimize this area. Since it is more likely that all places referenced in a text are close to each other (e.g., if we have two candidates for *Berlin*, namely *Berlin* in *Germany* and *Berlin* in the *United States*, and in the same text we have *Vienna* in *Austria* and two candidates for *Munich*, namely *Munich* in *Germany* and *Munich* in the *United States*, the candidates that form the smallest area in terms of a convex hull are, *Berlin* in *Germany*, *Vienna* in *Austria* and *Munich* in *Germany*).
- One sense: Assumes that if a place reference is made in a text, all the other place references with the same name are referring to the same place (i.e., resolve all place references sharing names with earlier resolved place references).

- Focus regions: If the source of the document has some geographic focus (e.g., the source of the document is a local newspaper of a certain city), all place references candidates that lie outside of this focus region are discarded. A geographic focus can also be computed with basis on the unambiguous references. Considering a set of unambiguous place references in the document, all the place reference candidates that lie outside this focus are discarded.

- Feature type disambiguation: If the place is referenced as having a certain relation to a place type (e.g., *city of X*, *lake Y near X*, etc.) then we should discard the place candidates that are know as not having that relation (e.g., having the place reference *city of Scotland* in a certain text drastically reduces the possibility that other references to *Scotland* are references to the country with the same name).

- Textual−Spatial Correlation: This heuristic assumes that the text where a place reference occurs should be similar to a textual description for the correct place being referenced (i.e., a text describing the correct place is more similar to the context where the place reference is inserted). Thus, disambiguation should be made to the candidate having the highest textual similarity (e.g., through the TF-IDF method introduced in Section 2.1.1) towards the textual document where the reference occurs.

- Comma Groups: This heuristic treats lists of place references with a certain similar characteristic (e.g., countries, states, cities, etc.). Authors of documents generally organize place references that share a common characteristic in a group (e.g., *Washington, Texas, Nevada* and *California*, are all states of the United States), and thus disambiguation should be made accordingly .

Besides the context rules and the prominence heuristics, there are others which do not fall into these categories. An example is the following rule:

- Edit distance: When querying a gazetteer with a place name, we can retrieve place references for which the name is not equal to the one queried (e.g., by making a query on the GeoNames[1] gazetteer for the name *Greece*, one can also obtain *Le Grès*, a populated place in France). We should disambiguate to the candidate whose name is more similar to the place reference.

Using a set of seven heuristic rules, Lieberman *et al.* (2010) built a system that tried to capture the geographic clues within a news article, that are made by the author of the article. This work is based on the assumption that the authors of news articles leave linguistic contextual clues

[1] http://www.geonames.org/

for the readers to understand the place references, and that the authors also bare in mind the reader's own place lexicons (e.g., a reader of a local newspaper from *Columbia* would know that a reference to *Amsterdam* would refer to *Amsterdam* in *Columbia*, and not *Amsterdam* in the *Netherlands*). The place lexicon is constructed for each source of news documents by collecting the locations that are more frequent in the set of documents, and by measuring the proximity of those places. Note that the local lexicons used by the authors can only be used in local news articles. The heuristic rules used by the authors are the following:

$H_1$**:** Dateline place references: In the beginning of the articles, the authors tend to locate the incident of the news. This information is thus particularly important in the disambiguation of the following references, and the authors propose to start by disambiguating dateline place references through heuristics $H_4$, $H_5$ and $H_6$.

$H_2$**:** Anchor place references: The authors of the articles tend to leave cue word patterns to identify the place where the news took place, i.e., the target region (e.g., *4 miles of Athens, Texas*). The authors propose to disambiguate anchor place references through heuristics $H_1$, $H_4$, $H_5$, $H_6$ and resolve the place references that are proximate to the target region.

$H_3$**:** Comma group: Treats place references with a certain similar characteristic, like being country names, using heuristics $H_6$, $H_5$ and using geographic proximity.

$H_4$**:** Location container: The authors of articles tend to use location containers to better describe the location (e.g., Lisbon, Portugal). The authors propose to query the gazetteer to find a pair that satisfies the reference, this way disambiguating the place reference.

$H_5$**:** Local lexicon: The authors propose that if a place reference exists in the local lexicon, then it should be disambiguated using the local lexicon.

$H_6$**:** Global lexicon: The authors propose that if a place reference exists in the global lexicon, then it should be resolved using the global lexicon (i.e., a lexicon of places that are prominent enough for a reader to recognize them independently of his location, like country names).

$H_7$**:** One sense: Assuming that a place reference does not have several interpretations throughout a document, the authors propose to resolve all place references sharing names with earlier resolved place references.

Based on the fact that journalistic documents are supposed to be read in a sequential manner, the authors use a sequential procedure based on the heuristic rules described above. These rules are applied by the rank order presented in the list of heuristics. To disambiguate place references, the procedure first finds the the location of where the event of the news occurred (i.e., $H_1$). Next,

the procedure looks for place references that denote some relative geography and tries to resolve them using $H_2$. All the place references near the target location are disambiguated as well. All listings of several place references are treated uniformly using $H_3$. Finally, all place references that were not disambiguated are resolved by using $H_4$, $H_5$, $H_6$, and $H_7$. The technique used by the authors achieved $F_1$ scores of 0.79 using an evaluation corpus of 428 news articles.

Other authors have proposed to use machine learning techniques to combine heuristic rules to disambiguate place references. Martins *et al.* (2010) used a regression model based on SVMs. The disambiguation process is done by querying a gazetteer (in their work, the authors used Geonames) for each place reference found in the text. The top ten results from the gazetteer are considered as candidates and, for each, the following features are computed to represent the candidate disambiguations:

- Levenshtein distance between the place reference and the candidate.
- The population count of the candidate.
- The number of alternative names for the candidate in the gazetteer.
- The metric distance between the geospatial coordinates of the candidate and the closest candidate for any other place reference found in the text.
- The areas of the convex and concave hull, computed from the centroid coordinates of the candidate, and from the coordinates of all the other candidates for other place references recognized in the same text.

To train the regression model, the authors compute the geospatial distance between each candidate and the true disambiguation. The regression model uses a training set $X$, corresponding to a set of $m$ pairs $x_i = \{f_{i,1}, ..., f_{i,m}, d_i\}$, where $f_{i,j}$ is the $j_th$ input feature of a given example $i$ and $d_i$ is the corresponding geospatial distance. The goal of the learning algorithm is to find a function $reg(x)$ which, for a given input $x'_i = f_{i,1}, ..., f_{i,m}$ produces a distance result that is as close as possible to the target $d_i$ for every $x'_i$. The authors achieved an average $F_1$ score of 0.66 in an English corpus of news documents. Compared to other commercial state-of-the-art systems like Yahoo! Placemaker[1] and Metacarta GeoTagger[2], which achieved an average of $F_1$ score of 0.57 and 0.42 respectively, over the same collection, the proposed technique achieved a significant improvement.

[1] http://developer.yahoo.com/geo/placemaker
[2] http://www.metacarta.com/

### 2.2.2 Document Georeferencing

The relationship between language and geography has long been a topic of interest to linguists (Johnstone, 2010). Many studies have, for instance, shown that geography has an impact on the relationship between vocabulary terms and semantic classes. For instance, the term *football*, in the United States, refers to the particular sport of American football. However, in regions such as Europe, the term *football* is usually associated to different sports (e.g., soccer or, less frequently, rugby football). Terms such as *beach* or *snow* are also more likely to be associated to particular locations. In my MSc thesis, I am particularly interested in seeing if vocabulary terms and textual contents in general can be used to predict the geographical locations of documents. Georeferencing documents as a whole has many possible applications, namely for analyzing the distribution of opinions, topics, or other types of textual constructs.

Eisenstein *et al.* (2010) investigated the dialectal differences and the variations in regional interests over Twitter users, using a collection of georeferenced *tweets* and probabilistic models. These authors tried to georeference USA-based Twitter users with basis on their *tweet* content, concatenating all the *tweets* for each single user and using mixtures of Gaussian distributions to model the locations of the Twitter users.

Adams & Janowicz (2012) studied the relationship between topics present in documents and their geographic distribution. While most work in the area of geographic information extraction and retrieval relies on geographic keywords such as place names, Adams and Janowicz proposed an approach that only uses non-geographic expressions, in order to study if ordinary textual terms are also good predictions of geographic locations. The proposed technique first uses Latent Dirichlet Allocation (LDA) to discover latent topics present in a collection of documents. LDA is essentially an unsupervised generative method for modeling documents as a probabilistic mixtures of topics, which are in turn modeled as a probability distribution over a word vocabulary (Blei *et al.*, 2003). After fitting the LDA model, the authors use Kernel Density Estimation (KDE) to interpolate a density surface, corresponding to the entire world, for each of the LDA topics (Carlos *et al.*, 2010). Noticing that each document can be seen as a mixture of topics, the authors use map algebra operations to combine the density surfaces from each topic, finally assigning documents to the location having the highest density.

Anastácio *et al.* (2010) surveyed heuristic approaches to assign documents to geographic scopes, based on recognizing place references in the documents and afterwards combining the recognized references. The authors specifically compared approaches based on (i) the occurrence frequency for the references, (ii) the spatial overlap between bounding boxes associated to the

references, (iii) hierarchical containment between the references, using a taxonomy of administrative divisions, and (iv) graph-propagation methods using again a taxonomy of administrative divisions. Experiments with a collection of Web pages from the Open Directory Project showed that hierarchical containment achieved very good results.

Wing & Baldridge (2011), in a very similar study to the one that is reported in the Section 3.2 of this dissertation, compared different supervised approaches for automatically identifying the location of documents, expressed as latitude/longitude coordinates. The approach proposed by the authors is based on statistical models derived from a large corpus of already georeferenced documents, such as Wikipedia. The authors divide the Earth's surface through a geodesic grid of squared cells of equal degree. This approach produces variable-sized regions that shrink latitudinally, becoming progressively smaller and more elongated the closer they get towards the poles. However, the authors claim that since most populated regions are close to the equator, this error will not be relevant. Having a discrete representation of the Earth, the authors find a multinomial distribution over words, for the vocabulary for each cell in the grid. Given a grid $G$ with cells $c_i$ and a vocabulary $V$ with words $w_j$ this distribution is represented by $\theta_{c_ij} = P(w_j|c_i)$. A distribution of words for each document $d$ is represented by $\theta_{d_kj} = P(w_j|d_k)$. These distributions are computed straightforwardly from training documents. In order to classify unseen documents, the authors experimented with the Kullback-Leibler (KL) divergence and a Naive Bayes classifier. Using the KL divergence between each unseen document distribution and each cell distribution, the authors compute the following quantity:

$$LK(P|Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \tag{2.31}$$

In the formula, P is the distribution of words for a given document, and Q is the distribution of words for a given cell. This quantity measures how good Q is as an encoding for P, and the smaller this value is, the better. The cell that is more similar to the given unseen document is the one that provides the best encoding.

More recently, the authors published a new study, in which the Earth's surface is represented using an adaptive grid. This grid is constructed using a k-d tree structure (Bentley, 1975), which groups nearby points into buckets. The partition given by the k-d tree provides a finer granularity in dense regions, and a coarser granularity in sparse regions. Using this new technique the authors achieved better results than in their previous work, at the same time also improving computational efficiency.

The work reported in Section 3.2 of this dissertation is very similar to that of Wing and Baldridge, but I propose to use (i) a different scheme for partitioning the set of documents into bins of equal

area, according to their geospatial locations, (ii) a different language modeling approach for classifying documents according to the most similar bins, (iii) a hierarchical decomposition approach for improving the computational performance of the classification method, and (iv) different post-processing techniques for assigning the geospatial coordinates with basis on the obtained classification scores.

### 2.2.3 Opinion Based Document Classification

Opinion Mining is different from traditional Information Extraction and document classification tasks, as it presents several particularly hard challenges that require creative solutions. Suppose we want to classify a document by counting words that are associated with positive or negative opinions. Choosing one such list of words is not a trivial task, as words may have different interpretations according to context. Pang *et al.* (2002), in a study with movie reviews, tested the difficulty that humans have in discerning good words to use for the classification of sentiments. Using two lists of words made by two people, and a classifier that simply used word counts, the authors achieved an accuracy of 0.58 and 0.64. Using a list of words obtained from statistical methods achieved an accuracy of 0.69. They also obtained better results in ties (i.e., in documents that could be classified as either positive or negative). This work allows us to understand the subjectivity of this area, and the difficulty in distinguishing between positive and negative sentiment words.

Even if we could make an accurate list of all the words or phrases with a positive or negative sentiment, a sentiment can not always be identified using keywords or key phrases. For example, although opinion words like *horrible* or *good* can be considered as standard sentiment identifiers, we can easily find positive or negative opinion expressions that do not use any of the typical opinion words. For example, the sentence *how could anyone sit through this movie* clearly points to a negative opinion about a certain movie, without using any of the typical opinion words or phrases (Pang *et al.*, 2002).

Another major challenge relates to the fact that although the notion of positive and negative is fairly consistent across multiple domains, some expressions can encode different sentiments when used in different domains. For example, while *go read the book* may have a positive connotation in the context of a book review, it may have a negative connotation if the expression is used in the context of a movie review (Pang & Lee, 2008). Also note that in opinion based document classification, having a negation in a sentence can completely change the class of that sentence, unlike in the typical document classification scenario where sentences like *this car*

*does not have five doors* do not change the fact that the text is talking about cars.

There are several applications related to this new area of text mining. Examples include the classification of movie reviews, the analysis of political opinions, or the analysis of opinions from customers regarding products.

To classify an opinion, we can resort to a two-point opinion classification scale (i.e., the classification of an opinion as either positive or negative). This type of classification is called Binary Polarity Classification of Opinion. Related work and known methods for binary opinion classification are presented in Section 2.2.3.1. Since opinions can have different intensities, and can sometimes be considered neutral, it may also be interesting to analyze the level of positivity of an opinion. This type of problem falls in the area of Multi-Scalar Opinion Classification. Work related to this type of classification, and known methods, can be found in Section 2.2.3.2.

#### 2.2.3.1 Binary Polarity Classification of Opinions

The usage of lexicon-based classifiers has been explored by many researchers working on opinion mining. This has proven to be a simple method with promising results. The basic idea is to use a list of positive and negative words to classify opinions in a given text. For instance Hu & Liu (2004) created a lexicon-based opinion mining system to classify client reviews, summarizing them by product feature. The method developed by the authors consists of three main steps. First, the frequent features commented by the users are identified. This is achieved by first tagging the reviews on the database with part-of-speech (POS) tags, using the NLProcessor linguistic parser[1] (Malkovsky & Subbotin, 2000). After the tagging is done, an association rule miner is run on the noun phrases, and the false frequent features (i.e., features identified by the previous step, which are in fact not real features) are discarded by applying two pruning methods, in which the first checks features that contain at least two words and removes those that are likely to be meaningless, and the second removes redundant features.

The second step concerns with identifying opinion sentences and their polarity. The opinion sentences are the sentences that have one or more frequent features and one or more opinion bearing adjectives (i.e., opinion words). Using a list of positive and negative adjectives, the opinion words are classified as having a positive or negative connotation. This list was made through an adjective seed list, for which the sentiment orientation is known, and through WordNet, an online dictionary containing word synonyms and antonyms (Miller & Fellbaum, 2007). The basic

[1] http://www.infogistics.com/textanalysis.html

idea is that synonym adjectives share the same orientation, and antonyms have the opposite orientation. When classifying an opinion word with an unknown sentiment, the procedure searches in WordNet for synonyms or antonyms with a known sentiment orientation. The new opinion word is added to the seed list with the orientation of its synonym, or with the opposite orientation of the antonym. The seed list grows during the process of classification. The classification of the product feature is made by counting the number of positive and negative opinion words. In case of a tie, the polarity is classified as the sentiment of the effective opinion word (i.e., the closest opinion word of the feature). The algorithm also treats negation words. If a negation word appears near an opinion word (according to a defined threshold), the polarity of the opinion word is inverted. Finally, the summarization is made by counting the number of positive and negative features of each product review. To test their method, the authors extracted 500 reviews of five electronics products from the sites `Amazon.com` and `Cnet.com`. They achieved an average sentence orientation accuracy of 0.84, showing promising results for this type of technique.

Pang *et al.* (2002) tested whether the classification of opinions could be addressed as a special case of the classical topical classification of texts. To this end, three classification models were tested, namely Naive Bayes, Maxium Entropy and Support Vector Machines. In order to treat negations, for each word or phrase occurring between a negative word and the next punctuation mark, a new word was added to the dictionary (e.g., *i don't like this movie* is treated as *i don't NOT_like this movie* ). Various tests were made with different sets of features (i.e., unigrams, unigrams + bigrams, POS + unigrams, adjectives, the most 2633 frequent unigrams, and unigrams + position). The authors found that each model behaves better or worse depending on the type of features being used, but all models obtained better results using feature presence in the document, rather than the feature frequency in the document. In the case of Naive Bayes, the best result corresponded to an accuracy of 0.82 using unigrams with POS tags. The best result with Maxium Entropy models corresponded an accuracy of 0.81 using the 2633 most frequent unigrams in the corpus of documents. Finally, the Support Vector Machine model presented the best results, with a maximum of 0.83 of accuracy when using unigram features.

Turney (2002) presented a particularly innovative idea, capable of achieving good results in some contexts, using an unsupervised learning algorithm. The algorithm starts by assigning POS tags to the text, and discards phrases that do not have adjectives or adverbs. The basic idea is to extract adjectives from sentences to know the sentiment and the next word to give the context (depending on the context, an adjective can have a positive or negative connotation). The second step is to use the Pointwise Mutual Information - Information Retrieval (PMI-IR) algorithm to estimate the polarity of a phrase. This algorithm uses mutual information as a measure of the strength for the semantic association between two words. Specifically, it compares the strength of

the semantic association between the extracted phrase with the words *excellent* and *poor*, using statistics taken from a search engine, in order to classify the sentence with a positive or negative connotation. The Pointwise Mutual Information between two phrases $w_1$ and $w_2$ is:

$$PMI(w_1, w_2) = log_2 \left( \frac{p(w_1 \cap w_2)}{p(w_1)p(w_2)} \right) \tag{2.32}$$

In the formula, $p(w_1 \cap w_2)$ is the probability that $w_1$ and $w_2$ co-occur. When the probability of $w_1$ and $w_2$ co-occurring is independent, we have that the probability of co-occurrence is given by the product of $p(w_1)$ and $p(w_2)$. Therefore, PMI is a measure of statistical dependency between the phrases. The Semantic Orientation (SO) of a phrase is given by the formula:

$$SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor") \tag{2.33}$$

The semantic orientation of the phrase is positive when the phrase is more associated with the word *excellent*, and negative when the phrase is more associated with the word *poor*. PMI-IR estimates PMI by making queries to a search engine and counting the number of hits. The author used the AltaVista Search Engine because, at the time, it supported the NEAR operator for retrieving documents where two search terms appear together. Using the search engine, the SO of a phrase can then be estimated by:

$$SO(phrase) = log_2 \left( \frac{hits(phrase\ NEAR\ excellent) hits(poor)}{hits(phrase\ NEAR\ poor) hits(excellent)} \right) \tag{2.34}$$

Notice that Equation 2.34 can be derived from Equations 2.32 and 2.33 with some minor algebraic manipulations. The tests on this method were conducted using reviews extracted from the site `Epinions.com`. The reviews were taken from the following domains: automotive, banks, movies and travel destinations. The method achieved an accuracy of 0.84 in reviews about cars, and an accuracy of 0.67 in reviews about movies. The author concludes that it is difficult to classify movie reviews due to the subjectivity of the domain. For example, the phrase *evil guy* is found to have a negative connotation by the method tested, but having an evil character is not necessarily negative in the context of a movie review.

#### 2.2.3.2 Multi Scalar Opinion Classification

Instead of just classifying a document as having either a positive or negative sentiment, one can

think that it would be interesting to measure the positivity of a document. In their work, Pang & Lee (2005) addressed the rating-inference problem, attempting to classify documents using a three class scale or a four class scale. For this purpose, they used a meta-algorithm based on metric-labeling. The basic idea of this method is to use an initial label preference function that gives an estimation of how to label the items. Using this function, it is possible to compute the label of a new item according to the similarity towards other labeled items. The initial label preference function $\Pi(x, l)$ can be obtained using many methods. The authors used two special types of SVMs, namely a One vs All classifier and a regression model.

Let $d$ be the metric distance between labels, and let $nn_k(x)$ be the $k$ nearest labels of item $x$, according to a similarity function $sim(x, y)$. Then, the problem of labeling the target item can be solved by minimizing the formula:

$$\sum_{x \epsilon test} \left[ -\Pi(x, l) + \alpha \sum_{y \epsilon nn_k(x)} f(d(l_x, l_y)) sim(x, y) \right] \tag{2.35}$$

The parameter $\alpha$ represents a trade-off and/or scaling parameter, while $f$ is a monotonically increasing function. The similarity function used in the work of Pang & Lee (2005) was obtained using the positive-sentence percentage for items $x$ and $y$ (i.e., the percentage of positive sentences in a review). The reason why the authors used this similarity measure was due to a test they made using reviews from four authors extracted from the site `rottentomatoes.com`. They noticed that all four authors tend to have higher positive-sentence percentages (PSPs) for reviews with higher positivity assignments. Therefore, the similarity function used is $sim(x, y) = cos(\overrightarrow{PSP(x)}, \overrightarrow{PSP(y)})$, where $\overrightarrow{PSP(x)}$ is the vector (PSP(x), 1- PSP(x)).

To test this method, a corpus of 5006 reviews from four different authors was extracted from the site `rottentomatoes.com`. When using a three point scale, the authors achieved an average accuracy of 0.66 using metric labeling with a One vs All SVM as the initial labeling preference function. They obtained an accuracy of 0.62 using the regression model as the initial labeling preference function. When using a four point scale, the method achieved an average accuracy of 0.52 using metric labeling with a One vs All SVM, and an accuracy of 0.54 using metric labeling with the regression model. The authors tested the metric-labeling method, using One vs All SVMs or the regression model, and compared the results with those obtained by just using SVM classifiers. All the tests indicated that metric labeling was the better classifier, except in the four point scale test, where the regression model and metric labeling were almost equivalent.

### 2.2.4 Summary

This chapter presented the concepts necessary to understand the work that has been made in the context of my MSc thesis, together with the related work in which I based my work. Section 2.1 presented fundamental concepts and existing techniques for resolving various tasks in Information Extraction and Retrieval. Specifically, we have seen that:

- Many techniques exist in order to retrieve relevant documents based on keywords. The term frequency-inverse document frequency heuristic can be used to represent documents and queries as vectors of terms, assigning more importance to terms that are very frequent in a certain document, but that are rare in the rest of the documents of the corpus. Documents and queries can be compared on the basis of those vectors.
- In order to address the task of Named Entity Recognition, which relates to classifying words in a text into certain categories, such as names of persons, locations or organizations, it is common to use techniques such as Hidden Markov Models or Conditional Random Fields, which are probabilistic models that can be learned from data.
- The task of classifying documents relates to finding the best category, among a list of predefined categories to be assigned to a given document. Language Model Classifiers appear as a natural approach, as each category can be modeled as a language model which returns the probability of having generated a given document. Support Vector Machines are also commonly used in text categorization problems.
- Maps are the primary mechanism for summarizing and communicating geographically related information. Thematic maps are suited to represent geographical related information, and can be classified into point maps (e.g. proportional symbol maps), area maps (e.g., choropleth maps) and line symbol maps (e.g. flow maps).
- Several evaluation metrics exist to measure the effectiveness of document classification or document retrieval systems, examples being precision and recall. However, these two metrics may not be enough, since a system can have a high recall at the cost of a low precision, and vice-versa. $F_1$ is a commonly used metric that relates both precision and recall. In classification problems, it is also common to use the accuracy metric, which is simply the fraction of correctly classified documents.

Section 2.2 presented previous works which are related to the subject of this MSc thesis. We have specifically seen that:

- Most work done so far in the area of GIR focused in the task of place reference resolution, where one first starts by identifying the place references present in a textual document, and afterwards disambiguates these references into their actual location on the Earth's surface.
- More recently, some researchers have started to explore approaches for georeferencing entire documents, using the intuition that some vocabulary terms present in documents, in combination, can be used to accurately predict geographical locations.
- Opinion mining relates to the extraction of opinionated information from textual documents. The area presents several non-trivial challenges, since sentiments cannot always be identified using keywords, and since some expressions can have a positive or negative connotation depending on their usage context. Opinions can be classified using a two-point classification scheme (i.e., polarity of opinion), or with multi-point classification schemes (e.g., through a five star scale).

# Chapter 3

# Mapping Opinions from Georeferenced Documents

This section describes the most important contributions of my MSc thesis research, which tried to prove that through Information Extraction and Information Retrieval techniques it is possible to discover and map the geographical distribution of opinions, extracted from a collection of textual documents. In order to validate my hypothesis, I created a prototype system with three main modules, namely a document georeferencing module, a sentiment analysis module, and a map generation module. The following section presents an overview of the prototype system, together with a description of the most important techniques that are used in the different modules. Section 3.2 details the methods explored for the task of georeferencing documents. Section 3.3 describes the methods explored for the task of analyzing the opinions expressed in documents. Finally, Section 3.4 describes the methods used for representing the opinions extracted from the georeferenced documents in a thematic map. Section 3.5 summarized the contents of this chapter

## 3.1 Overview

The prototype system that was developed in the context of my MSc thesis is divided into three modules, as shown in Figure 3.6. The system first classifies documents according to their implicit location on the surface of the Earth. Afterwards, it classifies the general opinion expressed in the document, using either a two-point or a five-point opinion scale. Finally, using the location of each document, and the opinions that were also extracted from each document, the map generation

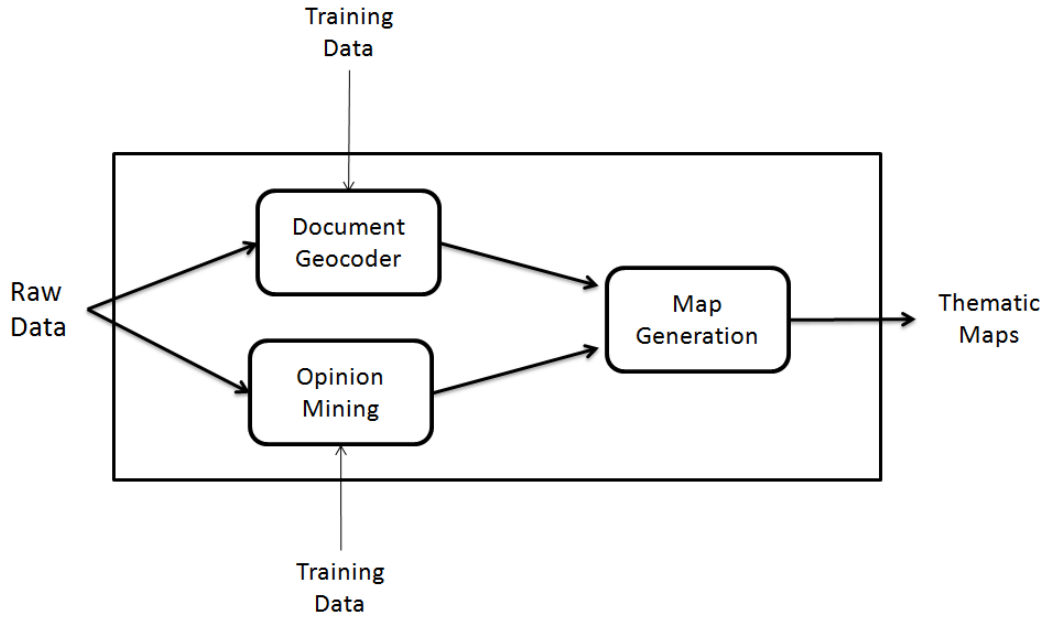



**Figure 3.6:** General architecture for the prototype system

module creates a thematic map based on a density surface computed for the opinions classes present in a document collection.

The document georeferencing module and the sentiment analysis module both classify documents according to pre-defined categories. The document georeferencing module divides the Earth's surface into roughly equally sized bins, and uses these bins as the categories for classification, finally returning for each unseen document the pair of coordinates of latitude and longitude that best describes the document, according to the assigned bin. The opinion mining module can perform two types of analysis, namely a two-point scale opinion analysis (i.e., determining the polarity of opinion), or a five-point scale opinion analysis. Both these modules are based on language model classifiers, implemented through the usage of the LingPipe package[1].

The map generation module was implemented through the usage of R[2], a software environment for statistical computing and graphics. Two specific R packages were used, namely ggplot2[3] for density estimation and general plotting, and maps[4] for map generation.

[1] http://alias-i.com/lingpipe
[2] http://www.r-project.org/
[3] http://had.co.nz/ggplot2/
[4] http://cran.r-project.org/web/packages/maps/index.html

### 3.1.1 LingPipe Language Model Classifiers

The LingPipe language model classifiers perform joint probability-based classification of textual documents into categories, based on either character-based or token-based language models (i.e., in the experiments that were performed, both these two different classification approaches were tested). The general idea is to build a language model $P(text|cat)$ for each category $cat$, afterwards building a multinomial distribution $P(cat)$ over the categories, and finally computing joint log probabilities for the classes according to Bayes's rule, yielding:

$$\log_2 P(cat, text) \propto \log_2 P(text|cat) + \log_2 P(cat) \tag{3.36}$$

In the formula, $P(text|cat)$ is the probability of seeing a given $text$ in the language model for a category $cat$, and $P(cat)$ is the marginal probability assigned by the multinomial distribution over the categories. The book by Carpenter & Baldwin (2011) has the complete details on the language models used for estimating $P(text|cat)$, and on the multinomial distribution $P(cat)$ over the categories. The multinomial distribution $P(cat)$ is basically estimated using a maximum a posteriori probability (MAP) estimate with additive (i.e., Dirichlet) priors.

In terms of the character-based or token-based language models, they are essentially generative language models based on the chain rule, which smooth estimates through linear interpolation with the next lower-order context models, and where there is a probability of $1.0$ to the sum of the probability of all sequences of a specified length.

## 3.2 Georeferencing Textual Documents

The proposed method for georeferencing textual documents uses only the raw text of the documents as evidence, relying on a discrete binned representation of the Earth's surface. The bins from this representation, corresponding to roughly equally-sized areas of the Earth, are initially associated to textual documents (i.e., all the documents from a training set that are known to refer to each particular bin are used). A compact representation is built from these georeferenced sets of documents, based on character-based or token-based language models, capturing their main statistical properties. New documents are then assigned to the most similar bin. Finally, each document is assigned to their respective coordinates of latitude and longitude, with basis on one of four different post-processing techniques.

The following section describes the approach used to represent the Earth's surface, Section 3.2.2

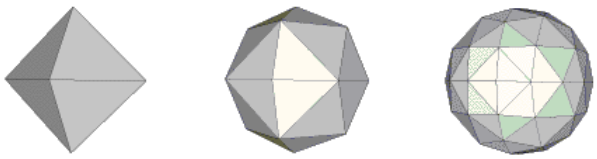


**Figure 3.7:** Decompositions of the Earth's surface for triangular meshes with resolutions of 0, 1 and 2.

describes the different post-processing techniques that were considered in order to assign coordinates to previously unseen documents. Finally, Section 3.2.3 details a hierarchical classification approach, which was used to improve the computational performance of the proposed method.

### 3.2.1 The Hierarchical Triangular Mesh

The proposed approach for representing the Earth's surface is based on discretizing space into a set of bins, allowing us to predict locations with standard approaches for discrete outcomes. However, unlike previous authors such as Serdyukov *et al.* (2009) or Wing & Baldridge (2011), which used a grid of squared cells of equal degree, the method proposed in this section is based on a Hierarchical Triangular Mesh[1] (Dutton, 1996; Szalay *et al.*, 2005). This strategy results in a triangular grid that roughly preserves an equal area for each bin, instead of variable-size regions that shrink latitudinally, becoming progressively smaller and elongated as they get closer towards the poles. Notice that this binned representation ignores all higher level semantic regions, such as states, countries or continents. Nonetheless, this is appropriate for the purpose of this work, since documents can be related to geographical regions that do not fit into an administrative division of the Earth's surface.

The Hierarchical Triangular Mesh (HTM) offers a multi-level recursive decomposition for a spherical approximation to the Earth's surface – see Figures 3.7 and 3.8, both adapted from original images at the HTM website. The decomposition starts at level zero with an octahedron and, as one projects the edges of the octahedron onto the sphere, it creates 8 spherical triangles, 4 on the Northern and 4 on the Southern hemispheres. Four of these triangles share a vertex at the pole, and the sides opposite to the pole form the Equator. Each of the 8 spherical triangles can be split into four smaller triangles by introducing new vertices at the midpoints of each side, and adding a great circle arc segment to connect the new vertices. This sub-division process can be

[1] `http://www.skyserver.org/htm/Old_default.aspx`

| Resolution | 4 | 6 | 8 | 10 |
|---|---|---|---|---|
| Total number of categories | 2,048 | 32768 | 524,288 | 8,388,608 |
| Average area of each bin ($km^2$) | 28,774.215 | 17,157.570 | 1,041.710 | 67.031 |

**Table 3.1:** Number of bins in the triangular mesh and their corresponding area.

repeated recursively, until we reach the desired level of resolution. The triangles in this mesh are the bins used in the representation of the Earth, and every triangle, at any resolution, is represented by a single numeric ID. For each location given by a pair of coordinates on the surface of the sphere, there is an ID representing the triangle, at a particular resolution, that contains the corresponding point.

Notice that the proposed representation scheme contains a parameter $k$ that controls the resolution, i.e. the area of the bins. Having course grained bins can lead to very rough estimates, but classification accuracy, with a thin-grained resolution, can also decrease substantially, due to insufficient data to build the language models associated to each bin. In the experiments that were conducted, this parameter ranged from $4$ to $10$, with $0$ corresponding to the first-level division. Table 3.1 presents the maximum number of bins that would be generated at each of the considered levels of resolution. The number of bins $n$ for a resolution $k$ is given by $n = 8 * 4^k$. Table 3.1 also shows the area, in squared Kilometers, corresponding to each bin.

### 3.2.2 Post-Processing for Assigning Geospatial Coordinates

With the HTM-based discrete representation for the Earth's surface, the LingPipe[1] package was used to build character-based or token-based language models, afterwards using these models for associating, to each bin, the probability of being the best class for a given novel document. In the conducted experiments, the character-based language models were based on sequences of 8 characters. As for the token-based language models, sequences of tokens with a 2-gram model were captured, and the models considered white-spaces and unknown tokens separately.

[1] `http://alias-i.com/lingpipe`

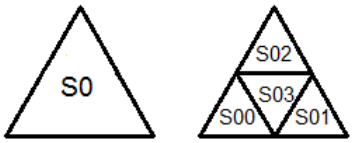



**Figure 3.8:** Recursive decomposition of the circular triangles used in the triangular mesh.

After having probabilities assigned to each of the bins from the representation of the Earth, latitude and longitude coordinates are computed with basis on the centroid coordinates for the most probable bin(s). In this particular stage, experiments with four different post-processing techniques were conducted. These are as follows:

1. Assign geospatial coordinates with basis on the centroid of the most probable bin.
2. Assign geospatial coordinates according to a weighted average of the centroid coordinates for all the possible bins, where the weights come from the probabilities assigned to each of the bins by the classifier.
3. Assign geospatial coordinates according to a weighted average of the centroid coordinates for the most probable bin and for its adjacent neighbors in the hierarchical triangular mesh, again using weights corresponding to the probabilities assigned to each of the bins.
4. Assign geospatial coordinates according to a weighted average of the coordinates associated to the $knn$ most similar documents in the training collection that are contained in the most probable bin.

Methods two and three, from the previous enumeration, require for the classifier to return calibrated probabilities, whereas the language modeling approach used in LingPipe is know to produce skewed and too extreme probability estimates. In the literature, there are many methods for calibrating classification probabilities by post-processing, but most of these methods are only defined for binary classification problems (Bella *et al.* (2009); Gebel & Weihs (2007)). In this particular multi-class problem, the values returned by the language model classifier were processed by using a sigmoid function of the form $(\sigma * score)/(\sigma - score + 1)$, where the $\sigma$ parameter controlling the gradient of the curve was adjusted empirically.

In what regards method four, the similarity between documents was computed according to the cosine similarity between document feature vectors, built with basis on term bigrams. In the experiments that were conducted, the $knn$ parameter ranged between $5$ and $20$.

### 3.2.3 Improving Performance Through Hierarchical Classification

Although language model classifiers can be used directly to assign documents to the most probable bins, they can be very inefficient in practice when considering a thin-grained resolution, due to the very large number of classes – see Table 3.1 – and due to the need for estimating, for each document, its probability of having been generated by the language model corresponding to each class. In order to solve this problem, I propose to use a hierarchical classification approach,

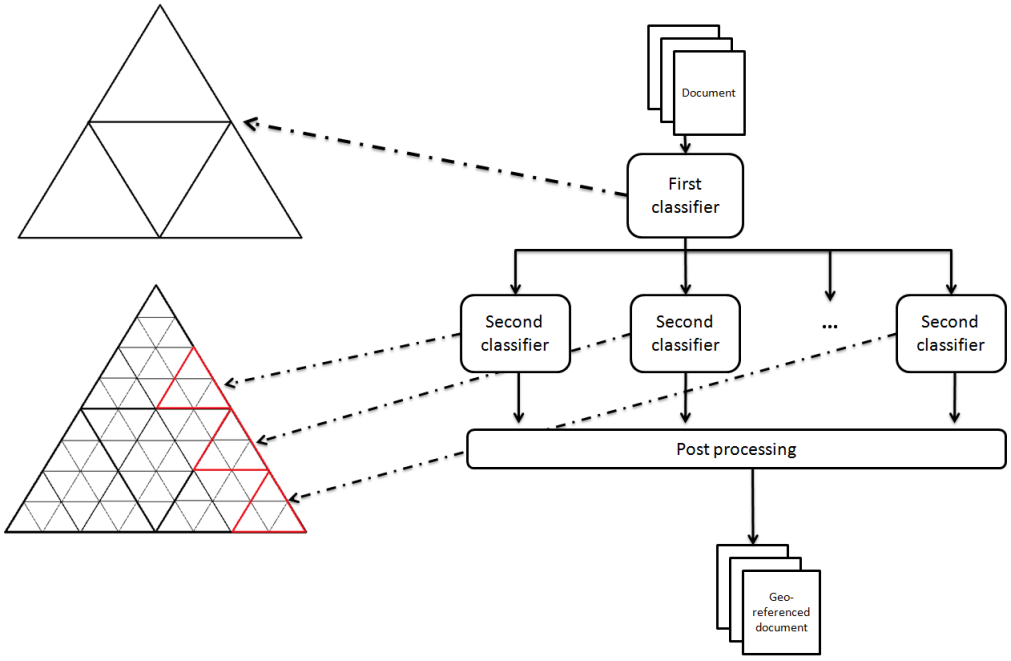



**Figure 3.9:** The hierarchical classifier for document georeferencing.

where instead of a single classifier considering all bins from a detailed triangular mesh encoding the Earth's surface, a hierarchy of classifiers with two levels is used, as demonstrated in Figure 3.9. The first level corresponds to a single classification model using bins from a coarse-grained division of the Earth, whereas the second level corresponds to different classifiers, one for each class from the first level, encoding different parts of the Earth with a thinner granularity. With this hierarchical scheme, classification can be made much more efficiently, as documents need to be evaluated with less language models.

I also propose a technique for reducing the number of classes in the second level classifiers. If a given bin from the decomposition of the Earth does not contain any training documents assigned to it, and if only one of its neighbouring bins in the mesh contains documents, then a single class from the hierarchical triangular mesh of the immediately smaller resolution is used, in order to represent this region in the classification model.

In a related previous work, Wing & Baldridge (2011) reported on very accurate results (i.e., a median prediction error of just 11.8 Kilometers, and a mean of 221 Kilometers) with a similar, but non-hierarchical classification approach, based on the Kullback-Leibler divergence between language models. However, these authors also claim that a full run with all their experiments (i.e., six different strategies) required about 4 months of computing time and about 10-16 GB of RAM, when run on a 64-bit Intel Xeon E5540 CPU. The hierarchical classification approach previously

described can substantially reduce the required computational effort. On similar hardware, and with a Wikipedia dump of approximately the same size, the full set of experiments reported on Section 4.1 run by the method that I propose took approximately 4 days to complete.

## 3.3 Mining Opinions from Text

Two approaches based on language model classifiers were considered for the task of mining opinions from textual documents, namely one using a two-point scale (i.e., polarity of opinion) and another using a multi-point scale, in this latter case considering a five-point scale. For both approaches, the LingPipe[1] package was used to build character-based and token-based language models, as described in Section 3.1.1. The considered classification approaches were based on the sentiment analysis tutorial described in the LingPipe Website[2], with some minor modifications. In the conducted experiments, the character-based language model classifier was based on sequences of 8 characters, and the token-based language model classifier was based on sequences of tokens with a 2-gram model, and with the white-spaces and unknown tokens modeled separately.

One important difference between the two-point scale opinion analysis scenario and the multi-point scale case, is that, in the second case, we can leverage the fact that nearby classes should be more similar than far-away classes, and similar data should receive similar labels. In order to model this idea, a post-processing method know as metric-labeling was used to smooth the results of the language model classifier, described in Section 2.2.3.2. The algorithm was adapted in order to suit the multi-point scale problem at hand, where the language model classifier was used as the initial label function, and the similarity function used was the cosine similarity between documents, resulting in the following equation:

$$\sum_{x \epsilon test} \left[ -LM(x,l) + \alpha \sum_{y \epsilon nn_k(x)} f(d(l_x, l_y)) CosineSim(x,y) \right] \tag{3.37}$$

In order to speed up both the training and the classification, a hierarchical classification approach was also developed. This approach is similar to the one presented in Section 3.2.3, and is also composed of two levels of classification. In the first level we have a single classifier that distinguishes between positive and negative documents. In the second level we have two classifiers, namely one that distinguishes levels of negativity, and a second classifier that distinguishes levels

[1] http://alias-i.com/lingpipe
[2] http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html

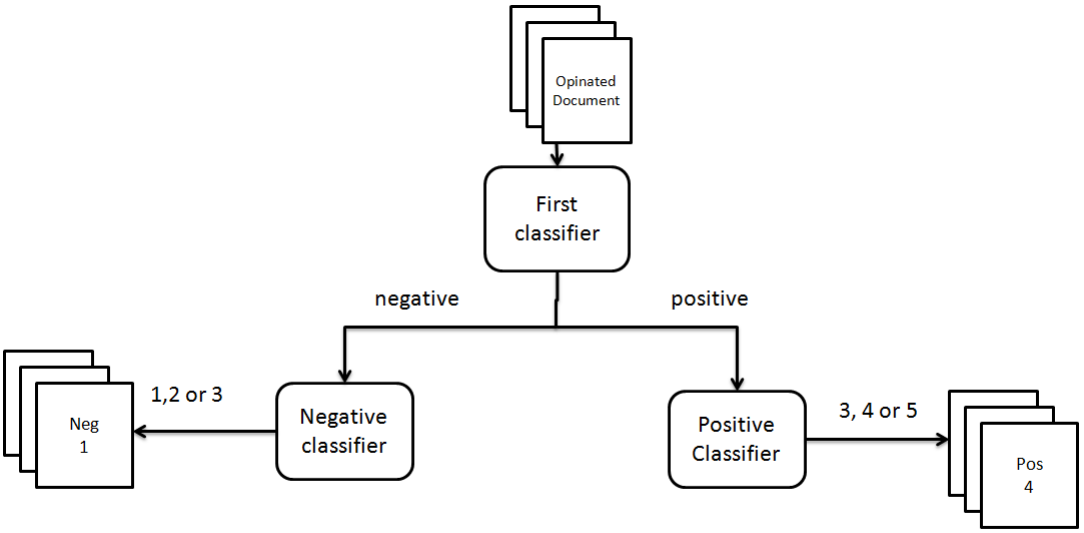



**Figure 3.10:** Composition of the hierarchical classifier for sentiment analysis.

of positivity, as shown in Figure 3.10. Category 3 from the five-point opinion scale was considered in both the positive and negative classifiers, since this category is usually associated with a neutral opinion.

## 3.4 Mapping the Extracted Opinions

To generate a thematic map portraying the geospatial distributions of opinions, towards a particular theme, I propose to leverage on a collection of documents related to the theme of interest, starting by using the methods that were described in the previous sections for (i) assigning each of the documents to geospatial coordinates of latitude and longitude, and (ii) assigning each of the documents to a particular opinion class. Afterwards, I used the geospatial coordinates of latitude and longitude to generate a density surface for each of the considered opinion classes. The density surfaces are generated through a popular approach for smoothing geospatial data called Kernel Density Estimation (KDE), which essentially offers a non-parametric way to estimate the probability density function of a random variable such as the number of observations relating to a particular location.

In brief, we have that KDE is a non-parametric method for extrapolating data points over an area of interest, without relying in fixed boundaries for aggregation. KDE fits a density surface over each data point, such that the surface is highest over the data point and zero outside an estimated bandwidth (i.e., a defined distance from the data point). The value for the density surface in a

particular point is given by the following formula:

$$f(x,y) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{d_i}{h}\right) \tag{3.38}$$

In the formula, $f(x,y)$ is the density at a given location $(x,y)$, $n$ is the number of data points available in a sample, $h$ is the bandwidth, $d_i$ is the geographical distance between data point $i$ and location $(x,y)$, and $K()$ is a kernel function which integrates to one. The KDE was implemented through the usage of the R package ggplot2, which estimates the bandwidth using a data-driven method proposed by Sheather & Jones (1991). In this implementation, the kernel function that is used corresponds to the Gaussian kernel, defined by:

$$K(d,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} \tag{3.39}$$

In the formula, $\sigma$ determines the width of the kernel. The generated density surfaces can be overlayed on a map for the direct exploration of the strength that a particular opinion class has according to different regions (i.e., they can be used as maps showing the number of cases reporting either positive or negative opinions in each particular region). However, in order to get an aggregated representation, I propose to leverage on map Algebra operations to combine the different density surfaces. By considering that positive documents have a positive density and negative documents have a negative density, overlaying two such surfaces will result in a positive or negative surface. By associating a particular color to the different density surfaces (e.g., red for the negative opinions, green for the positive ones, and white for neutral), and through the usage of an appropriate transparency level, we can produce aggregate maps that show the regions where opinions tend to be more positive or more negative. When using opinions in a two-point scale, we combine the two different surfaces, obtained through the KDE method, in order to form a continuous surface with values from $-1.0$ to $1.0$, according to the formula $KDE(positive) - KDE(negative)$. Thus, regions where there are more negative or positive opinions will be associated, respectively, to a negative or a positive value. Regions where there are few opinions, or where we have an equal proportion of positive and negative opinions, will be assigned to a value that is close to zero. A similar procedure is used for the case when we have opinions in a five-point scale, but we instead normalize the KDE values for each opinion class into the interval $[0; 0.4]$, and we then map each opinion class to an interval within the limit values of $-1.0$ and $1.0$. The opinions corresponding to a value of 3 in the three-point scale are assigned to the interval from $-0.2$ to $0.2$, and since documents expressing a neutral opinion are often closer to expressing a negative evaluation, we perform the mapping in way such that regions where

**Figure 3.11:** An example overlay of two density surfaces.

there's a higher density of neutral opinions, end up being associated to a value that is closer to $-0.2$. Figure 3.11 show an example map illustrating an overlay of two density surfaces.

## 3.5 Summary

This chapter presented the main contributions of my MSc research, outlining the general architecture for the prototype system that was developed, and detailing the methods explored for the task of georeferencing documents, the methods explored for the task of analyzing the opinions expressed in documents, and finally the methods used for representing the opinions extracted from the georeferenced documents in a thematic map.

I specifically proposed a method for georeferencing textual documents that uses a hierarchical classification approach based on language model classifiers, relying on a discrete representation of the Earth's surface called Hierarchical Triangular Mesh. This particular method has also been published as a paper at the Spanish Conference on Information Retrieval (CERI 2012) (Dias *et al.*, 2012). I also proposed four different post processing methods to assign coordinates to textual documents. These four post-processing methods were published in a technical report and submited in the Portuguese journal for the automatic processing of the Iberic languages (Linguamática).

I also proposed to analyze the opinions expressed over the documents using two approaches that are also based on language model classifiers, namely a two-point scale classification approach and a five-point scale classification approach. In the case of the second approach, a post-processing technique called metric-labeling was also considered to improve classification accuracy.

With the extracted information, one can create many types of thematic maps. In order to make sense of the extracted information, I proposed the creation of density maps, using the kernel density estimation technique in order to interpolate a surface between the data points corresponding to the coordinates of the georeferenced documents belonging to particular opinion classes.

# Chapter 4

# Validation Experiments

This chapter describes the experimental validation of the methods that were proposed for geocoding documents, for mining opinions from text, and for generating thematic maps. In order to measure the quality of the obtained results, and to validate my research hypothesis, two types of tests were conducted, namely tests at a module level (i.e., each module was tested individually), and tests at a system level. In order to validate the module for geocoding documents, a collection of georeferenced Wikipedia documents was used, comparing the coordinates that were automatically assigned, against the ones available in the collection. To validate the module for opinion mining, and map module, a collection of reviews extracted from the website `yelp.com` was used.

## 4.1 Evaluating Document Georeferencing

This section describes the experimental methodology used for comparing the proposed methods for georeferencing textual documents, afterwards discussing the obtained results. For the experiments reported for document geocoding, a sample of the articles from the English Wikipedia dump from 2012 was used. Included in this dump are a total of 4,080,270 articles, of which 430,032 were associated to latitude and longitude coordinates. Previous studies have already shown that Wikipedia articles are a well-suited source of textual contents for the purpose of evaluating document georeferencing methods (Overell, 2009; Wing & Baldridge, 2011).

The Wikipedia dump was processed in order to extract the raw text from the articles and for extracting the geospatial coordinates, using the software dmir wikipedia-parser[1], which is based

[1] `http://code.google.com/p/dmir-wiki-parser/`

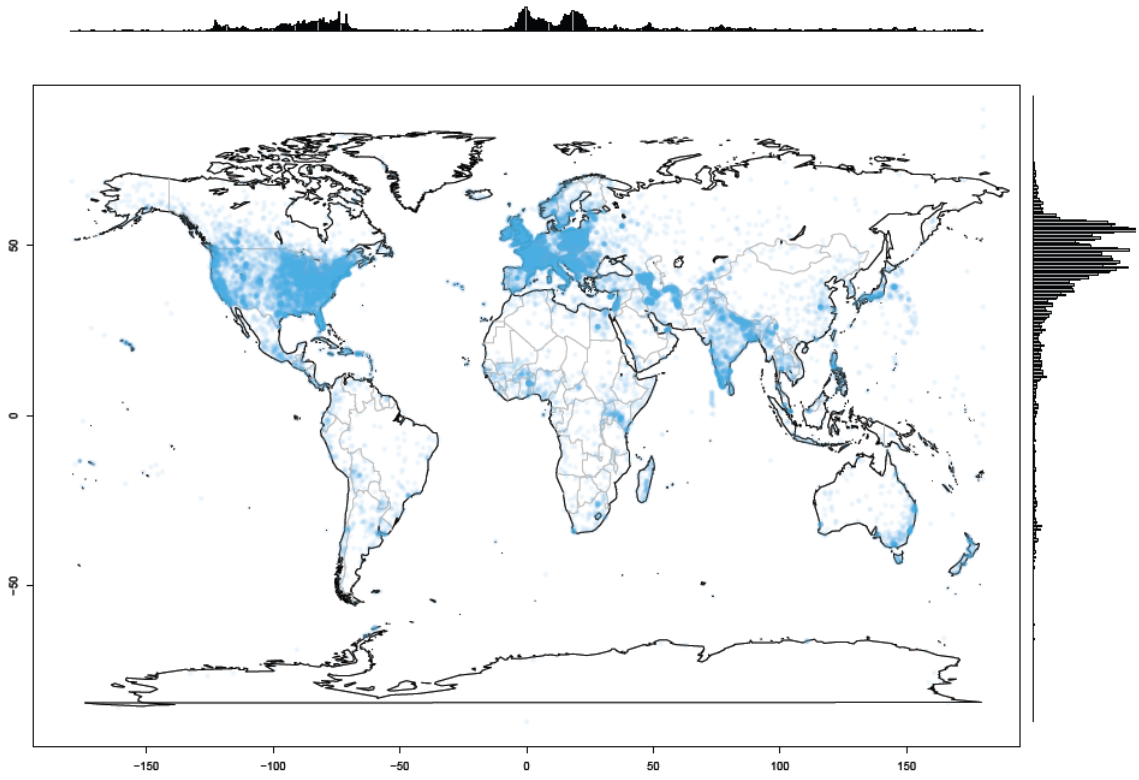



**Figure 4.12:** Geographic distribution for the Wikipedia documents.

on manually-defined patterns to capture some of the multiple templates and multiple formats for expressing latitude and longitude in Wikipedia. Considering a random order for the georeferenced articles, about 91% of the georeferenced articles that could be processed were used for model training (i.e., a total of 390,032 articles) and the other 9% were used for model validation (i.e., a total of 40,000 articles). Table 4.2 presents a statistical characterization for the considered dataset, while Figure 4.12 illustrates the geospatial distribution of the locations associated to the Wikipedia documents. Notice that some geographic regions (e.g, *North America* or *Europe*) are considerable more dense in terms of document associations than others (e.g, *Africa*). Moreover, oceans and other large masses of water are scarce in associations to Wikipedia documents. This implies that the number of classes that has to be considered by the model is much smaller than the theoretical number of classes given in Table 3.1. In the Wikipedia dataset that was used, there are a total number of 1,123 bins containing associations to documents at resolution level 4, and a total of, 8,320, 42,331 and 144,693 bins, respectively at resolutions 6, 8 and 10.

In order to get some insights into the hypothesis that general textual terms can be indicative of geographic locations, documents were first filtered according to their containment of particular terms, and then the corresponding coordinates were plotted on a map. Figure 4.13 shows the

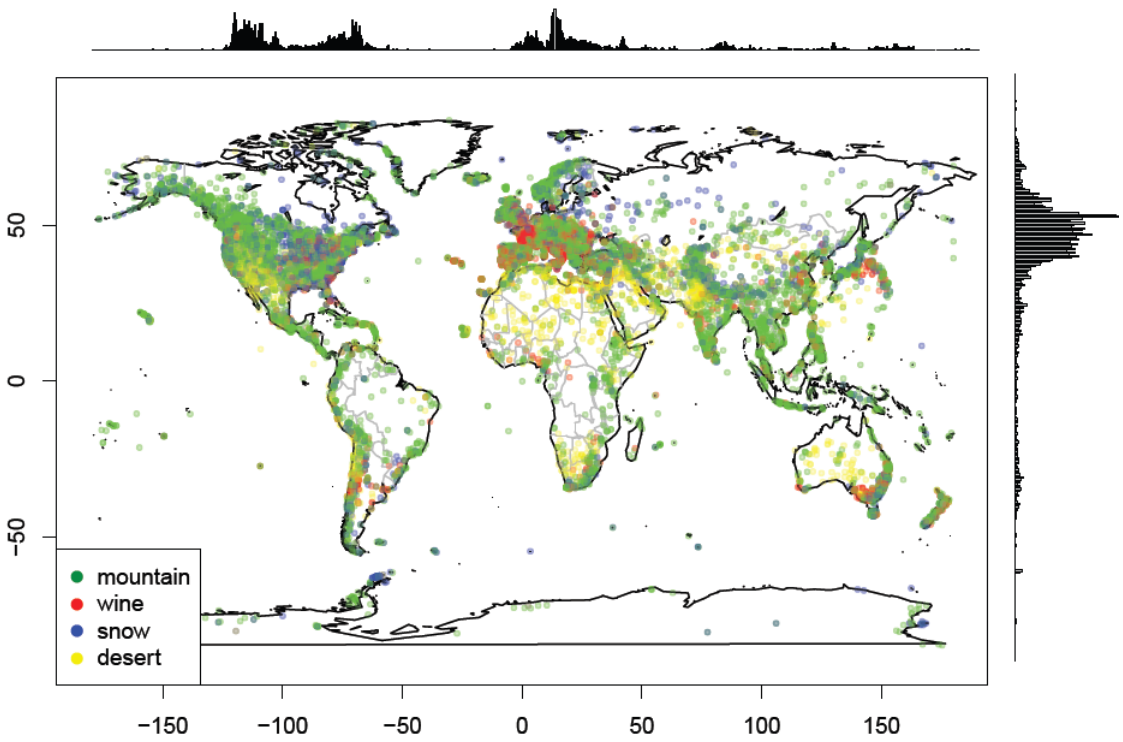



**Figure 4.13:** Geographic incidence of particular terms.

geographic incidence of four different textual terms, namely *mountain* using green dots, *wine* using red dots, *snow* using blue dots, and *desert* using yellow dots. The figure shows that these particular terms are more associated to the regions that would be expected (i.e., terms like *wine* are more associated to regions such as *France*, or terms like *desert* are more associated to *North Africa* or *Southwest United States*).

To get some insight that certain words are more associated to certain regions, I made an analysis of the most significant words of some regions against the rest of the World. In order to do this, I used a technique that evaluates the significance of phrases in one collection versus another.

| Statistic | Train | Test |
|---|---|---|
| Number of Documents | 390,032 | 40,000 |
| Number of Words | 160,508,876 | 16,696,639 |
| Average Words per Document | 411 | 417 |
| Standard Deviation of Words per Document | 74.202 | 231.705 |

**Table 4.2:** Characterization of the Wikipedia dataset.

| New York | Alps | Australia | Mediterranean |
|---|---|---|---|
| , Massachusetts | bar : | potato cod | Beni - |
| New Jersey | Communes of | Cape Leveque | - Abbes |
| , Connecticut | * Communes | Mount Isa | - Garonne |
| New York | France . | HMAS Sydney | - Atlantiques |
| Rhode Island | Czech Republic | Angas Downs | Béni Abbès |
| in Massachusetts | the Czech | Creal Reef | Garonne department |
| , Pennsylvania | ” bar | Cay ( | Naâma Province |
| Massachusetts . | a commune | Robbins Island | of Crete |
| , New | the Municipality | Tanami Gold | the Municipality |
| . Accessed | Municipality of | Bitter Victory | de Catalunya |

**Table 4.3:** Most significant bi-grams for certain regions.

The considered significant phrases are the ones that occur more often in a foreground corpus against a background corpus. To create the background model, I trained a language model with every Wikipedia document present in the collection, retaining the statistical characteristics of the World. For each region in analysis, I trained a language model with the documents enclosed in that region's bounding box. Table 4.3 presents the top ten more significant bi-gram of tokens present in four regions. We can see for instance that in the region of New York, the top eight more significant bi-grams are place references, in fact we can notice that most bi-grams are indeed place references, which shows that place references present in textual documents have a major impact in the language models decision. Another thing that we can notice is that some bi-grams associated to certain regions seem out of place (e.g., *Czech Republic* appears as one of the most significant bi-grams for the regions of the Alps, which clearly seems peculiar), this may be due to the fact that bounding boxes can contain other regions besides the one they are trying to enclose.

Using the Wikipedia dataset, experiments were conducted with classification models relying on bin sizes of varying granularity. Table 4.4 presents the obtained results for some of the different methods that were under study (i.e., all types of classifiers and the three first post-processing strategies, which did not use the similarity of neighbouring documents), together with the error values for each bin size. The prediction errors shown in Table 4.4 correspond to the distance in Kilometers, computed through Vincenty's formulae[1] (Vincenty, 1975), from the predicted locations to the locations given at the gold standard. The accuracy values correspond to the relative number of times that it was possible to assign documents to the correct bin (i.e., the bin where the document's true geospatial coordinates of latitude and longitude are contained). The $k1$ and $k2$ values correspond to the resolution for the Earth's representation, used at each level of the

[1] `http://en.wikipedia.org/wiki/Vincenty's_formulae`

| | Resolution | | Classifier Accuracy | | Geospatial Distance | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Centroid | | All Bins | | Neighbour Bins | |
| Method | k1 | k2 | 1st Level | 2nd Level | Average | Median | Average | Median | Average | Median |
| Character Models | 0 | 4 | **0.9609** | **0.8354** | 405.214 | 240.017 | 438.762 | 228.829 | 386.271 | 219.379 |
| | 1 | 6 | 0.9411 | 0.6669 | **254.846** | 62.846 | 282.874 | 71.119 | 257.741 | 65.551 |
| | 2 | 8 | 0.9283 | 0.3989 | 268.480 | **25.757** | 283.761 | 48.039 | 269.493 | 28.569 |
| | 3 | 10 | 0.8912 | 0.1615 | 281.669 | 30.405 | 287.909 | 51.595 | 281.755 | 30.464 |
| Token Models | 0 | 4 | 0.9444 | 0.5209 | 693.764 | 240.017 | 1544.543 | 228.829 | 691.899 | 219.379 |
| | 1 | 6 | 0.9103 | 0.4244 | 433.224 | 92.770 | 729.546 | 303.055 | 444.766 | 120.561 |
| | 2 | 8 | 0.8909 | 0.2747 | 455.025 | 44.621 | 572.406 | 191.522 | 457.858 | 50.349 |
| | 3 | 10 | 0.8297 | 0.1305 | 547.760 | 42.162 | 591.111 | 123.457 | 547.996 | 41.982 |

**Table 4.4:** The obtained results for document geocoding with different types of classifiers.

hierarchical classifier.

The results from Table 4.4 show that the method corresponding to the usage of character-based language models obtained the best results, with the best configuration having a prediction accuracy of approximately 0.40 in the task of finding the correct bin, while assigning documents to the correct geospatial coordinates had an error of 268 Kilometers on average. The documents that were assigned to the correct bin had an average distance towards the correct coordinates of $12$ Kilometers. The results from Table 4.4 also show that both the second and third post-processing technique, in which the coordinates are adjusted with basis on a weighted average with all the bins or the neighboring bins, does not improve the results over the baseline method in which the centroid coordinates for the most probable bin is used. I believe that this is due to the fact that the language model classifiers do not provide accurate and well-calibrated probability estimates. Even when using the score calibration technique based on a sigmoid function, the method still produces too extreme estimates.

Table 4.5 presents the obtained results for language-model classifiers based on character n-grams (i.e., the best performing method from the previous experiment), when using the fourth post-processing method, in which the coordinates of latitude and longitude are assigned through a weighted average between the centroid coordinate of the most probable bin, and the coordinates of the $knn$ most similar training documents contained within that same bin. The first column of Table 4.5 indicates the number of considered nearest neighbors, while the $k1$ and $k2$ values correspond to the resolution for the Earth's representation used at each level of the hierarchical classifier based on character $n$-grams. The results show that using the 5 most similar documents, provided the best results, with an average distance error of just 265 kilometers, and a median distance error of 22 kilometers.

A visualization of the results obtained with the best performing method can be seen in Figure 4.14, where the map represents the geospatial distribution for the predicted locations. The figure shows that errors are evenly distributed, and also that *Europe* and *North America* remain

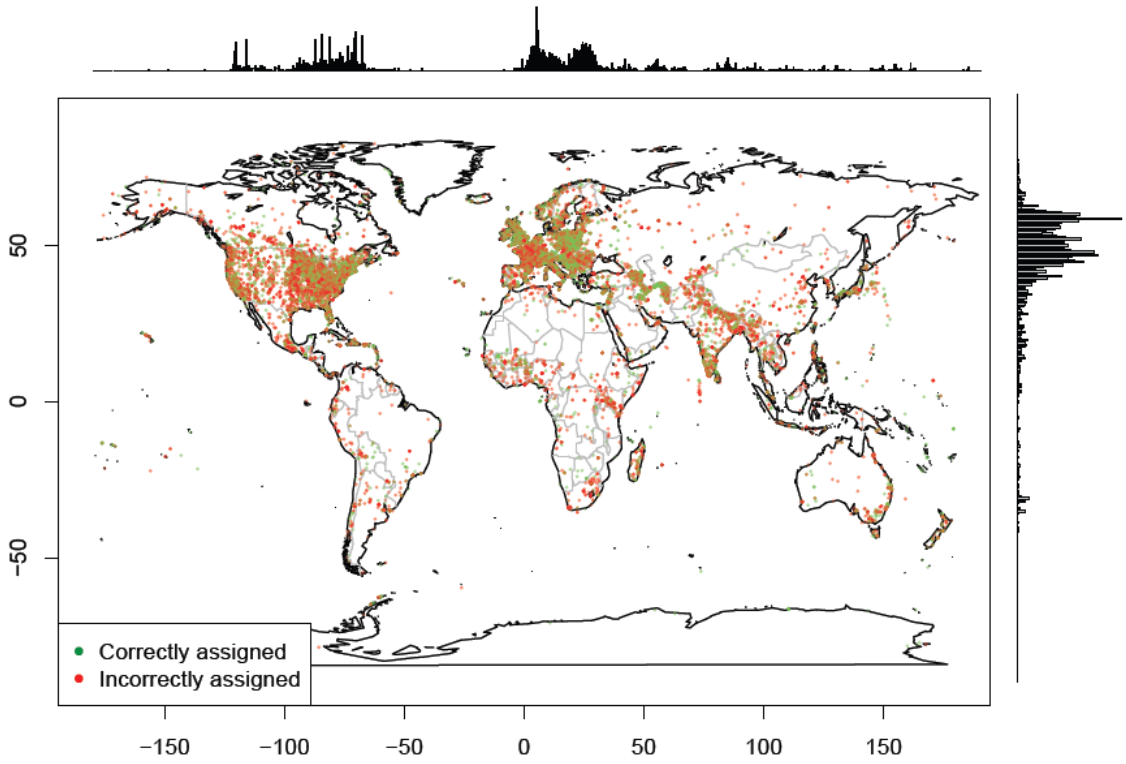



**Figure 4.14:** Estimated positions for the Wikipedia documents.

the regions with the highest density of documents.

Figure 4.15 illustrates the distribution for the errors produced by the character-based classifiers, in terms of the distance between the estimated coordinates and the true geospatial coordinates, when using the baseline method that assigns the centroid coordinates from the most probable bin, and when using the post-processing method that uses the coordinates from the 5 most similar documents. This figure plots the number of documents whose error (i.e., the distance towards the correct result) is greater or equal than a given value, using doubly logarithmic axes. Figure 4.15 shows that the proposed post-processing method based on the analysis of the most similar documents assigns coordinates to the majority of examples with a small error in terms of distance, with 31,634 documents having an error smaller than 100 Kilometers. Worse results are shown for the baseline method, with about 29,903 documents having an error smaller than 100 Kilometers.

Finally, in order to get some insight about the results, I made a correlation analysis between data and results. I analyzed two data properties against the geospatial distance error, namely the number of words per document and the number of place references present in each document.

| Nearest neighbours | Accuracy | Resolution k1 | Resolution k2 | Geospatial Distance Average | Geospatial Distance Median |
|---|---|---|---|---|---|
| 5 | 0.8354 | 0 | 4 | 289.750 | 90.515 |
| | 0.6669 | 1 | 6 | 235.702 | 34.005 |
| | 0.3989 | 2 | 8 | 265.734 | **22.315** |
| | 0.1615 | 3 | 10 | 281.442 | 30.209 |
| 10 | 0.8354 | 0 | 4 | 274.515 | 68.252 |
| | 0.6669 | 1 | 6 | 233.982 | 32.092 |
| | 0.3989 | 2 | 8 | 265.655 | 22.371 |
| | 0.1615 | 3 | 10 | 281.460 | 30.208 |
| 15 | 0.8354 | 0 | 4 | 271.045 | 64.008 |
| | 0.6669 | 1 | 6 | **233.928** | 32.243 |
| | 0.3989 | 2 | 8 | 265.744 | 22.480 |
| | 0.1615 | 3 | 10 | 281.464 | 30.172 |
| 20 | 0.8354 | 0 | 4 | 270.337 | 63.373 |
| | 0.6669 | 1 | 6 | 234.197 | 32.687 |
| | 0.3989 | 2 | 8 | 265.869 | 22.640 |
| | 0.1615 | 3 | 10 | 281.466 | 30.170 |

**Table 4.5:** Results for document geocoding with post-processing based on the $knn$ most similar documents.

Figure 4.16 illustrates the correlation between distance/references, distance/words and references/words. The top graphs shows the Spearman correlation, which represents the strength of correlation between two variables (Spearman, 1987). We can see that both the number of words and the number of place references have a weak correlation towards the geospatial distance error. Although, we can see from the scatter-plots that most documents with more than 5000 words have a small distance error. The same applies to the number of place references, where we can see that most documents with more than 100 place references have a small distance error.

## 4.2 Evaluating Opinion Mining

This section describes the experimental methodology used for evaluating the proposed methods for extracting opinions from textual documents, afterwards discussing the obtained results. For the experiments reported here, a sample of the reviews available from the website `yelp.com` was used (i.e., the data from the Yelp academic dataset[1]). Included in this collection are a total of 152,295 reviews. This collection includes reviews from clients of many business types, including restaurants, hotels and different kind of local shops, that are situated near some universities in the United States. The reviews in the Yelp dataset include ratings given by the users, using a five-point scale. Also included in this dataset are a pair of coordinates, latitude and longitude of

[1] `http://www.yelp.com/academic_dataset`

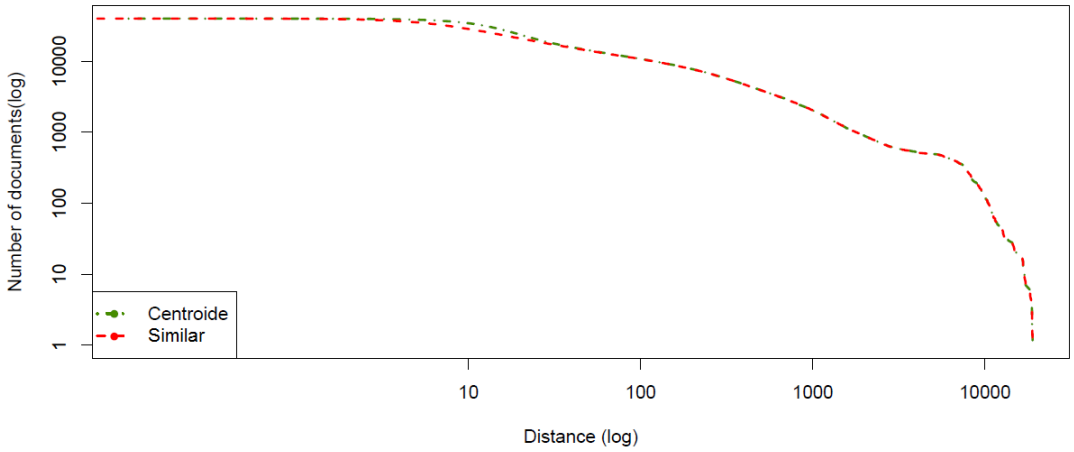



**Figure 4.15:** Distribution for the obtained errors, in terms of the geospatial distance towards the correct coordinates.

each business reviewed.

In a previous work, Cabral & Hortaçsu (2006) observed that users do not see neutral reviews, as between positive and negative, but rather they tend to see them as negative. In order to use this dataset in the polarity classification case, I considered that reviews having a rating of 3 stars (*i.e.*, neutral) would be considered as negative reviews, since most of these reviews had a mix between negative and positive phrases.

Considering a random order for the articles, about 74% of the reviews that could be processed were used for model training (i.e., a total of 112,295 reviews) and the other 26% were used for model validation (i.e., a total of 40,000 articles). Table 4.6 presents a statistical characterization for the considered dataset and the number of reviews per opinion category, where we can see that the number of negative cases is much smaller than the number of positive cases in both subsets.

Table 4.7 presents the results for the extraction of opinions with the two-point and five-point scales, for both the token based and the character based language models. We can see that character based language models achieved the best accuracy in all experiments. In the two-point scale case, an accuracy of 0.80 was achieved. The five-point scale case shows worse results, where the best method (i.e. using an hierarchical classification approach) only achieved an accuracy of 0.50. We can also see that the metric labeling approach only increased the accuracy by 0.07 points. In fact, of the 40,000 test cases, the metric labeling method only affected 323 cases. I believe this is due to the fact that the LingPipe language model classifiers do not

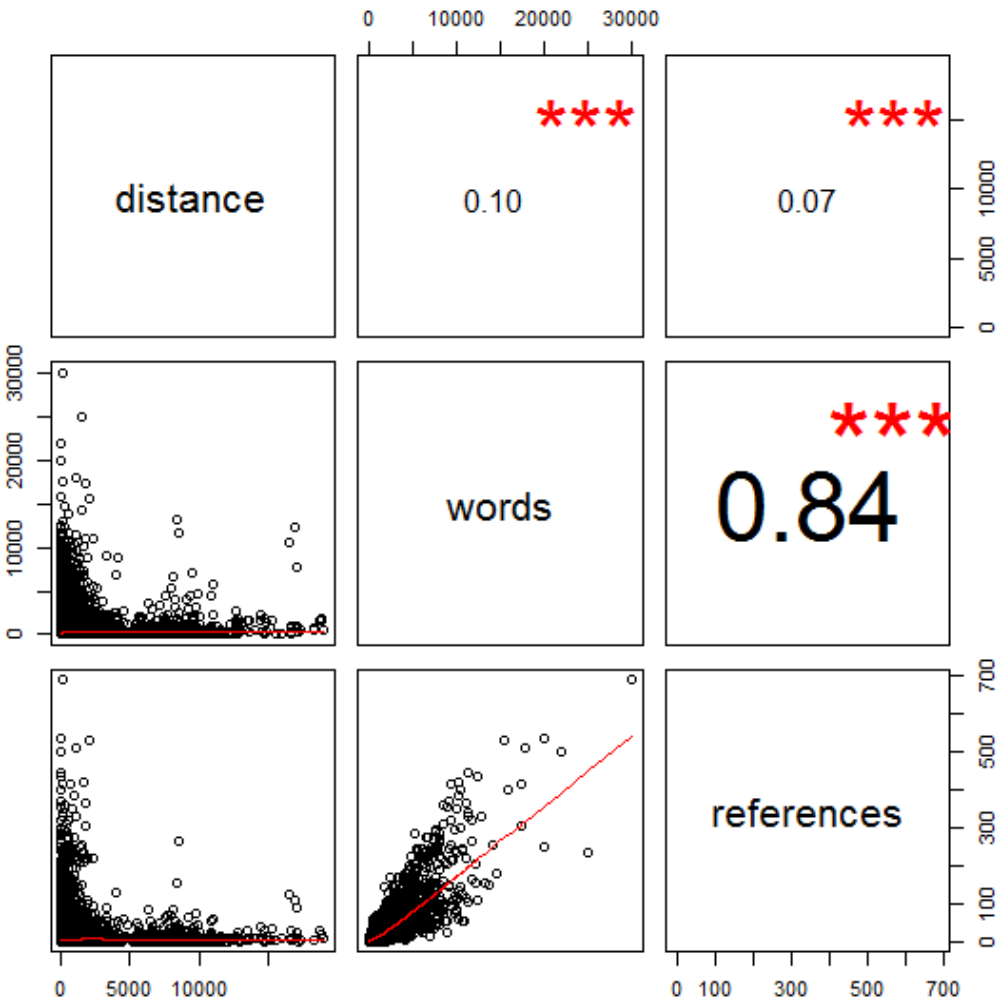



**Figure 4.16:** Correlation between data and results.

provide well-calibrated probability estimates.

Table 4.8 shows the precision achieved for each opinion class by the three methods tested for the five-point scale case. We can see that, for both the baseline and the metric labeling methods, opinion classes 2 and 3 present worse results, as was to be expected since these two categories have perhaps the most ambiguous language. Interestingly, the hierarchical method increases the precision for category 3 by approximately 0.10 points. This may be due to the fact that both second level classifiers have been trained with examples of category 3. Thus, even if the initial polarity classifier considers that a review that belongs to category 3 is positive, the second level classifier can still classify it as belonging to category 3.

| Statistic | Category | Train | Test |
|---|---|---|---|
| Number of Reviews | | 112,295 | 40,000 |
| Number of Words | | 17,038,778 | 6,041,821 |
| Average Words per Review | | 151 | 151 |
| Standard Deviation Words per Review | | 125.094 | 124.762 |
| Polarity | positive | 69,763 | 24,800 |
| | negative | 42,532 | 15,200 |
| 5 Star | 1 | 8,599 | 3,022 |
| | 2 | 11,872 | 4,219 |
| | 3 | 22,061 | 7,959 |
| | 4 | 38,615 | 13,664 |
| | 5 | 31,148 | 11,136 |

**Table 4.6:** Statistical characterization of the Yelp dataset.

| | | Multi-Scale | | |
|---|---|---|---|---|
| Model | Polarity | Normal | Metric Labeling | Hierarchical |
| Character Based | 0.8030 | 0.4940 | 0.4947 | 0.5008 |
| Token Based | 0.7577 | 0.4458 | 0.4461 | 0.4488 |

**Table 4.7:** The obtained results in terms of accuracy for multi-scale sentiment analysis.

## 4.3 Evaluating the Mapping of Opinions

In order to evaluate the generation of thematic maps portraying the geographic distribution of opinions, I again relied on the Yelp academic dataset. Different experiments were made with the Yelp dataset, separately accessing the issues related to the portrayal of opinions, from the issues associated to geospatial positional accuracy. Different maps were thus generated, namely:

- Maps using the ground-truth data available from the Yelp dataset, namely the review ratings in a scale from 1 to 5, and the geospatial coordinates for the local business being reviewed. These maps correspond to the types of representations that one would produce when using perfectly accurate text mining components, capable of exactly discovering the geospatial coordinates of documents, and their expressed opinions.
- Maps using the ground truth geospatial coordinates available from the Yelp dataset, and using the opinion polarity information derived from the proposed opinion mining method, either when considering a two-point opinion scale or a five-point opinion scale.
- Maps using the ground truth opinion polarity information from the Yelp dataset, and using the geospatial coordinates of latitude and longitude assigned by the proposed document geocoding method, in its most accurate configuration.

| Category | Normal | Metric Labeling | Hierarchical |
|---|---|---|---|
| 1 | 0.4173 | 0.4146 | 0.4302 |
| 2 | 0.2076 | 0.2079 | 0.2495 |
| 3 | 0.3118 | 0.3126 | 0.4194 |
| 4 | 0.6570 | 0.6603 | 0.6101 |
| 5 | 0.5530 | 0.5518 | 0.5392 |

**Table 4.8:** The obtained precision for each category for character based language models.

| 1st Level Precision | 2nd Level Precision | Average Distance | Median |
|---|---|---|---|
| 0.3163 | 0.0404 | 4154.017 | 3657.000 |

**Table 4.9:** The obtained results for the estimated positions of the Yelp dataset.

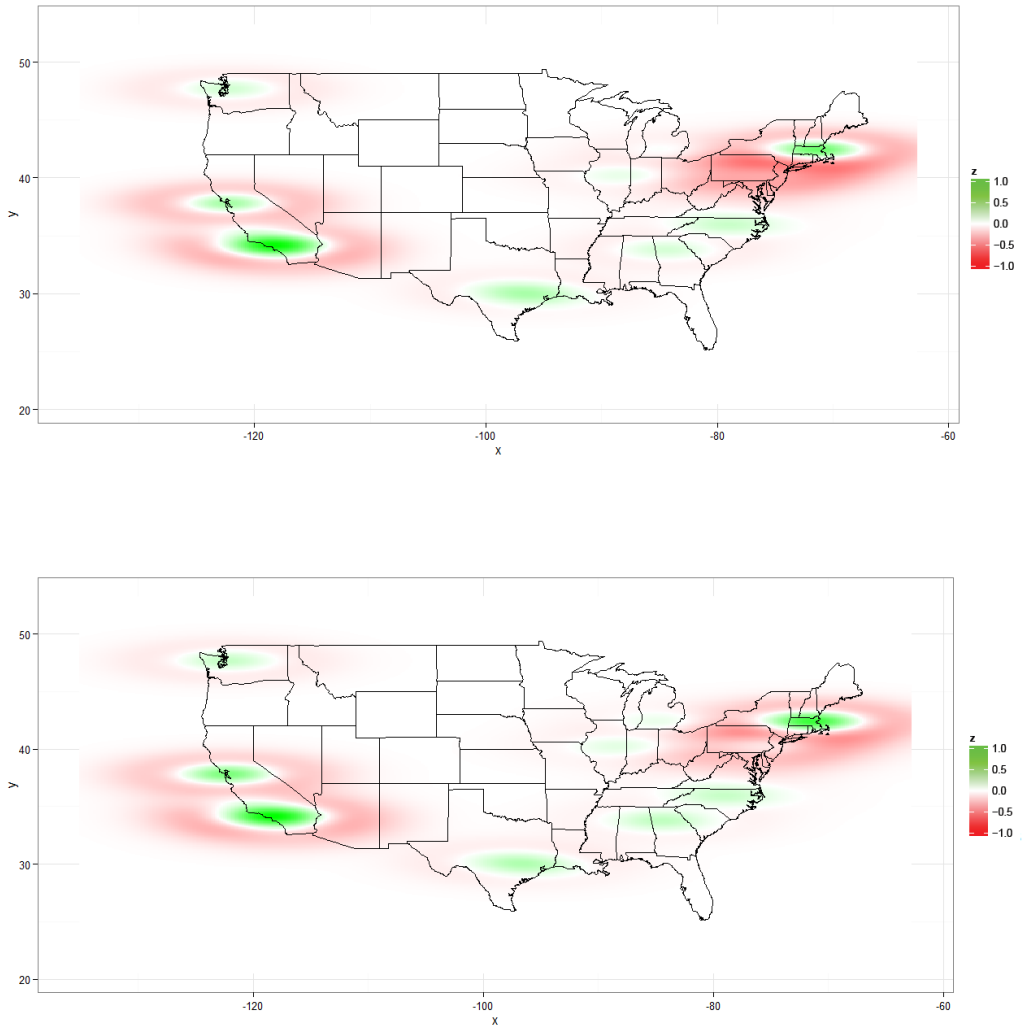
- Maps using only information derived automatically, when processing the Yelp collection with the proposed text mining methods, in their most accurate configurations.

The maps produced from methods 2, 3 and 4, from the previous enumeration, were compared against the map produced by method one.

Maps produced by method 1 and 2 from the previous enumeration are shown in figure 4.17, where the upper map represents the geographical distribution of opinions, using both real coordinates and opinions for each document. The bottom map represents the geographical distribution of opinions when estimating opinions, using the hierarchical classification approach for the five-point scale classification. We can see that the map with estimated opinions, closely resembles the real map, which proves that the proposed method for opinion mining suits the problem.

Figure 4.18 presents the estimated geographical distribution of opinions, where the upper map has the real opinion classifications, whereas the bottom map has both estimated coordinates and estimated opinion classes. We can see in the bottom figure that there are little density areas, clearly the estimated opinions nullify each other, as explained in Section 3.4. We can also notice in the mini world map that the georeference method classified some documents as belonging to the United Kingdom, probably because of some relation between equal city names in both the USA and the UK (e.g., like Cambridge or Oxford). In order to measure the quality of the estimated documents distribution of the Yelp collection, I analyzed the average distance error between the estimated position and the actual position of each review. Table 4.9 presents the results produced by evaluating the Yelp dataset, using the best configuration of the document georeferencing module. We can see that the results are not very accurate, with an average distance error of 4154 Kilometers and an accuracy of just 0.04. This is to be expected, since the

**Figure 4.17:** Thematic maps portraying the real geographic distribution of opinions

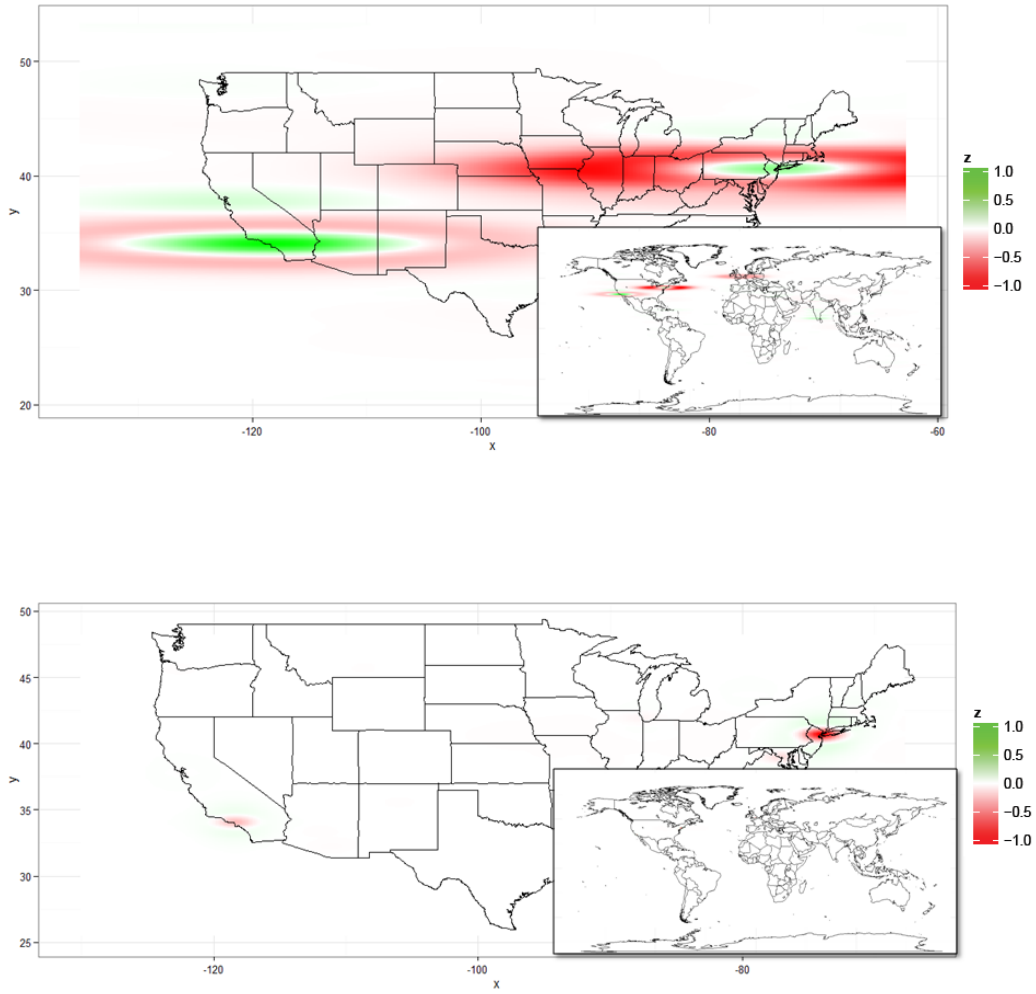



dataset was georeferenced using a model trained with the Wikipedia collection. The Wikipedia collection contains very descriptive documents, while the Yelp collection is composed by short reviews with an average of 151 words per document. Furthermore these reviews contain many opinion based words, while the Wikipedia documents nearly has none.

## 4.4 Summary

This chapter described the experimental validation for the techniques proposed in my MSc thesis. In order to test the methods for document georeferencing, which were described in Section

**Figure 4.18:** Thematic maps portraying the estimated geographic distribution of opinions




3.2, I used a collection of georeferenced Wikipedia documents. The experiments proved that character-based language models can achieve very good results when compared to previous work, such as the one presented by Wing & Baldridge (2011). Furthermore, the hierarchical classification approach can substantially reduce the computational effort. In the conducted experiments, using classifiers based on character-based language models, together with the post-processing method based on the analysis of the $knn$ most similar documents achieved the best results, with an average distance error of 265 Kilometers, and a median error of 22 Kilometers.

To test the proposed methods for opinion mining, a collection of reviews extracted from `yelp.com` was used. The experiments showed that character-based language models achieved better

results than token-based language models. In the two-point scale case, an accuracy of 0.80 was achieved. In the five-point scale case, the hierarchical classification method achieved the best results, with an accuracy of 0.50.

In what regards the generation of thematic maps, using the information extracted from documents, I experimented with the collection of reviews extracted from the Yelp dataset. We saw that using the Kernel density estimation to produce density maps, can indeed produce an accurate representation of the geographical distribution of opinions. However, estimating the distribution of opinions using the georeferencing model, trained with the Wikipedia dataset, proved to be a hard task, due to the differences between the two types of collections, where clearly the Yelp collection is composed of short opinionated reviews.

# Chapter 5

# Conclusions and Future Work

This dissertation presented the research work that was conducted in the context of my MSc thesis.

This report described different methods for assigning documents to a pair of geospatial coordinates of latitude and longitude that best summarizes their contents. It also shown one possible application for georeferenced information, namely finding and representing the geographical distribution of opinions. I studied different methods to analyze the overall opinion expressed in textual documents, using this information, I studied techniques to build thematic maps portraying the incidence of certain classes of opinions, in different geographic areas.

In what concerns georeferencing textual documents, I have shown that the automatic identification of the geospatial location of a document, based only on its text, can be performed with high accuracy, using simple supervised methods based on language modeling, and using a discrete binned representation of the Earth's surface, based on a hierarchical triangular mesh. The proposed method is simple to implement, and both training and testing can be easily parallelized.

In order to extract the overall opinion expressed in georeferenced documents, I evaluated methods for mining opinions using a two-point opinion scale and a five-point opinion scale scheme. I have shown that the automatic mining of opinions can be performed with high accuracy for the two-point scale case, using a simple method based on character-based language model classifiers. Worse results are shown for the five-point scale case, where the best performing method is based on a hierarchical language model classifier.

Using geographical information and opinions extracted from textual documents, I have shown that it is possible to represent the geographical distribution of opinions through thematic density maps. Estimating the geographical distribution of the Yelp collection proved to be a hard task,

given the non-descriptive short nature of the reviews, still the proposed method for representing the distribution of opinions, presents an effective way to summarize the data.

## 5.1 Main Contributions

Through a series of experiments, I have shown that *through Information Extraction and Information Retrieval techniques it is possible to find the geographical distribution of opinions, from a given collection of textual documents, and we can later represent this distribution in a thematic map*.

In order to validate the above hypothesis I built a prototype system composed of three main modules, namely a module for georeferencing textual documents, another module for mining opinions expressed in textual documents, and finally a module to generate thematic maps, representing the geographical distribution of opinions.

Through experiments with the first module, I have shown that the automatic identification of the geospatial location of a document, based only on its text, can be performed with high accuracy, using simple supervised methods, and using a discrete binned representation of the Earth's surface based on a hierarchical triangular mesh. The proposed method is simple to implement, and both training and testing can be easily parallelized. More specifically, my work provided the following main contributions:

- I proved that language model classifiers are suited to model the problem of extracting the geographical location that best describes documents, using only the raw text as evidence.
- I experimented with token-based and character-based language models. The results of those experiments proved that character based language models achieve better results, when evaluated over a collection of georeferenced Wikipedia documents, with the best configuration yielding an accuracy of 0.40, an average distance error of 268 Kilometers and a median error of 25 Kilometers.
- I developed a hierarchical classification approach, based on language model classifiers. This approach improves the computational performance, while yielding the same accuracy.
- I developed three post-processing methods, to make the location estimation more accurate. Using an analysis of the most similar documents produced the best results, with the best configuration achieving an average distance error of 265 kilometers and a median error of 22 kilometers.

In order to make available the research done in the area of geographical information retrieval, the proposed method for georeferencing textual documents as been published in the Spanish Conference on Information Retrieval (Dias *et al.*, 2012). More recently I also submitted a follow-up article to the Portuguese journal for the automatic processing of the Iberic languages (Linguamática), that includes the three post-processing methods to assign geospatial coordinates to textual documents. In this article a collection of Spanish and Portuguese Wikipedia documents was also considered, producing slightly worse results than the ones achieved with the English collection.

The prototype's source code has been shared in Google Code[1], in order to make it available to other researchers that work in the same field, together with an online demonstrator[2]

To analyze opinions present in documents, I used a classification approach, based on language models. I explored techniques to analyze the polarity of opinions and to analyze opinions according to a multi-point scale. Specific contributions of my work include:

- I experimented with token-based and character-based language model classifiers. Character-based language models have proven to be better in the conducted experiments, achieving 0.80 accuracy for the polarity classification case and 0.49 in the multi-point scale case.
- I adapted a method called metric labeling, in order to correct the classifications made by the model in the five-point scale case. The method did not produce the desired results, achieving almost the same accuracy as the baseline method. In the 40000 test cases, only 323 were corrected by the method. This is due to the fact that, LingPipe language models return non-calibrated probabilities, returning too extreme results.
- I created an hierarchical classification method similar to the one described in Section 3.2, in order to classify opinions in a five-point scale scheme. This method reduces the computational effort necessary to process both training and classification, moreover it also improved the accuracy to 0.5.

In order to represent the geographical distribution of opinions expressed in a collection of textual documents, I explored techniques to create density maps and I specifically :

- Applied a technique known as Kernel density estimation, in order to interpolate a density surface for each opinion class.
- Proposed a specific map algebra operation in order to add different opinion density surfaces into an overall opinion surface.

[1] `http://code.google.com/p/document-geocoder/`
[2] `https://appengine.google.com/dashboard/nonedeployed?app_id=s~lm-geocoder`

## 5.2 Future Work

In my MSc thesis dissertation we have seen automated techniques for assigning documents to opinion classes and to assign the geospatial coordinates that best summarize their contents. Using this information, we have seen that it is possible to build thematic maps that represent the geographic distribution of opinions. Although this technique proved effective, there are many possible techniques that would be interesting to investigate, in order improve the effectiveness of (i) finding the distribution of documents, (ii) analyzing the opinion expressed in textual documents, and (iii) representing the geographical distribution of opinions, based on thematic maps. Also, there are many possible applications for the methods proposed in my MSc thesis.

Using the techniques presented in Section 3.2, we could create a model that classifies textual documents according to the most probable period of time that best describes it. Also using the techniques described for georeferencing textual documents, and the techniques to create thematic maps it would be interesting to explore techniques to analyze documents within a geographical/temporal context. Where we could analyze the evolution of a certain topic in time (*e.g.* the evolution of opinions towards a certain theme in the World).

It should be noticed that the proposed classification approach, based on language models, does not provide accurate and well-calibrated probability estimates for the different classes involved in the problem, instead focusing only on the simpler task of predicting which class is the most likely. For future work, instead of just experimenting with an heuristic score calibration method based on post-processing, it would be interesting to experiment with other classification approaches for assigning documents to the most likely bin(s), for instance through maximum entropy models. It would also be interesting to experiment with maximum entropy models using either expectation constraints specifying affinities between words and labels (Druck *et al.*, 2008), or with posterior regularization (Ganchev *et al.*, 2010), leveraging the fact that the presence of the words corresponding to place names is a strong indicator for the document belonging to a particular class.

Also regarding place names, it is important to notice that although identifying a single location for an entire document can provide a convenient way for connecting texts with locations, useful for many different applications, many other applications could benefit from the complete resolution of place references in textual documents (Leidner, 2007). The probability distributions over bins, provided by the method described in Section 3.2, can for instance be used to define a document-level prior for the resolution of individual place names.

Having accurate and well-calibrated probability estimates for the different classes involved in the problem, should improve the post-processing methods that use these probabilities, such as the ones described for document georeferencing, described in Section 3.2.2. It would also be

interesting to explore the use of a meta-algorithm called metric labeling described in Section 2.2.3.2 and used in this MSc thesis in sentiment analysis. This algorithm post-processes the results given by a classifier (Pang & Lee, 2005) and tries to correct the probability estimates for each class so that similar documents receive similar labels.

It would also be interesting to explore the geographical distribution of topics, and study whether they are suited to model documents into their position in the Earth's surface. In an offline stage, this approach for georeferencing textual documents would perform the following steps:

- Start by pre-processing a collection of Wikipedia documents, in order to extract the full-text from each page, the links existing towards other pages, and the geospatial coordinates of latitude and longitude that are explicitly associated to some of the pages. These data will latter be used in the construction of our document geocoding models.
- Train a generative topic model from the contents of the Wikipedia collection, capable of representing documents as a mixture of possible topics. It would be interesting to experiment with the well-known Lattent Dirichlet Allocation model, as well as with an adapted version commonly refered to as Linked-LDA, which accounts with linkage information between the documents in addition to word co-occurences. Both these models are parametric, in the sense that they require for the user to specify the number of topics $k$.
- Take a subset of the Wikipedia documents that are explicitly associated to geospatial coordinates, and use these documents to fit a separate geospatial probability surface for each of the $k$ topics, using the latitude, the longitude and the topic probabilities associated to each document. The geospatial probability surface is obtained by interpolating the available data through the spherical interpolation methods for geospatial data available in the Spherekit toolkit.

For geocoding a new document, we could proceed as follows:

- Start by using the topic model for estimating the probability distribution over topics that characterizes the document, taking into account either only its textual contents in the case of the LDA model, or the textual contents together with the links towards Wikipedia documents (in the case of the Linked-LDA model). Notice that for documents not containing links to Wikipedia pages, we could also rely on entity linking methods for associating the entities referenced in the documents to the corresponding Wikipedia entries. Entity linking in text has indeed been receiving a significant interest over the last few years, with many different methods being proposed and evaluated in the context of the TAC Knowledge Base Population task.

- Use map Algebra operations to combing the geospatial probability surfaces previously generated for each topic, using the document's probability distribution over topics to weight the contribution of the different surfaces.
- The geospatial probability surface resulting from the combination represents the likelihood of having the textual document associated to different regions of the Earth. We could either use this surface directly as the geocoding result, or we could take the most probable point in the surface and return the corresponding coordinates of latitude and longitude.

Although the technique used for summing the density surfaces of each opinion class, proved to be effective, it can fail to convey information if the density surfaces from the opinion classes nullify each other. In future work, it would be interesting to study other techniques to represent several different density surfaces. I would also like to explore other types of thematic maps, such as choropleth maps, which I also think would fairly represent the sort of geographical information presented in my work.

Finally, I hope that the work produced during my thesis research will provide a basis for future works, towards finding the relationship between natural language and the geographical analysis of documents, in order to automatically translate a tangle of textual data into important information.

# Bibliography


ADAMS, B. & JANOWICZ, K. (2012). On the geo-indicativeness of non-georeferenced text. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*.

AMITAY, E., HAR'EL, N., SIVAN, R. & SOFFER, A. (2004). Web-a-where: geotagging web content. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

ANASTÁCIO, I., MARTINS, B. & CALADO, P. (2010). A comparison of different approaches for assigning geographic scopes to documents. In *Proceedings of the 1st Simpósio de Informática*.

BELLA, A., FERRI, C., HERNÁNDEZ-ORALLO, J. & RAMÍREZ-QUINTANA, M. (2009). Similarity-binning averaging: A generalisation of binning calibration. In *Intelligent Data Engineering and Automated Learning*.

BENTLEY, J.L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*.

BLEI, D.M., NG, A.Y. & JORDAN, M.I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*.

BOSER, B.E., GUYON, I.M. & VAPNIK, V.N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*.

CABRAL, L. & HORTAÇSU, A. (2006). The dynamics of seller reputation: Theory and evidence from eBay. Working paper, downloaded version revised in March.

CARLOS, H., SHI, X., SARGENT, J., TANSKI, S. & BERKE, E. (2010). Density estimation and adaptive bandwidths: A primer for public health practitioners. *International Journal of Health Geographics*.

CARPENTER, B. & BALDWIN, B. (2011). *Natural language processing with LingPipe 4*. LingPipe Publishing.

CHEN, S.F. & GOODMAN, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*.

DIAS, D., ANASTÁCIO, I. & MARTINS, B. (2012). A language modeling approach for georeferencing textual documents. In *Proceedings of the 2nd Spanish Conference in Information Retrieval*.

DRUCK, G., MANN, G. & MCCALLUM, A. (2008). Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in Information Retrieval*.

DUGAD, R. & DESAI, U.B. (1996). A tutorial on hidden markov models. Tech. rep., Indian Institute of Technology, Bombay.

DUTTON, G. (1996). Encoding and handling geospatial data with hierarchical triangular meshes. In *Advances in GIS Research II*, Taylor and Francis.

EISENSTEIN, J., O'CONNOR, B., SMITH, N.A. & XING, E.P. (2010). A latent variable model for geographic lexical variation. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*.

GANCHEV, K., GRAÇA, J.A., GILLENWATER, J. & TASKAR, B. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*.

GEBEL, M. & WEIHS, C. (2007). Calibrating classifier scores into probabilities. In *Advances in Data Analysis*.

HU, M. & LIU, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

JENNY ROSE FINKEL, T.G. & MANNING, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

JOHNSTONE, B. (2010). Language and place. In *Cambridge Handbook of Sociolinguistics*, Cambridge University Press.

KROVETZ, R., DEANE, P. & MADNANI, N. (2011). The web is not a person, berners-lee is not an organization, and african-americans are not locations: An analysis of the performance of named-entity recognition. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*.

LEIDNER, J.L. (2007). *Toponym resolution in text: Annotation, evaluation and applications of spatial grounding of place names*. Ph.D. thesis, University of Edinburgh.

LI, J. & HEAP, A. (2008). *A review of spatial interpolation methods for environmental scientists*. Geoscience Australia.

LIEBERMAN, M.D. & SAMET, H. (2011). Multifaceted toponym recognition for streaming news. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information*.

LIEBERMAN, M.D., SAMET, H. & SANKARANARAYANAN, J. (2010). Geotagging with local lexicons to build indexes for textually-specified spatial data. *International Conference on Data Engineering*.

MALKOVSKY, M.G. & SUBBOTIN, A.V. (2000). Nl-processor and linguistic knowledge base in a speech recognition system. In *Proceedings of the 3rd International Workshop on Text, Speech and Dialogue*.

MANNING, C.D., RAGHAVAN, P. & SCHÜTZE, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

MARTINS, B., ANASTÁCIO, I. & CALADO, P. (2010). A machine learning approach for resolving place references in text. In *Geospatial Thinking*.

MILLER, G. & FELLBAUM, C. (2007). Wordnet then and now. *Language Resources and Evaluation*.

MOGUERZA, J.M. & MUNTILDEOZ, A. (2006). Support Vector Machines with applications. *Statistical Science*.

OVERELL, S. (2009). *Geographic Information Retrieval: Classification, disambiguation and modeling*. Ph.D. thesis, Imperial College London.

PANG, B. & LEE, L. (2005). Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

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

PANG, B., LEE, L. & VAITHYANATHAN, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language processing*.

PENG, F., SCHUURMANS, D. & WANG, S. (2003). Language and task independent text categorization with simple language models. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

RABINER, L.R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*.

SANDVIK, B. (2008). Using kml for thematic mapping. Tech. rep., University of Edinburgh.

SCHMID, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

SERDYUKOV, P., MURDOCK, V. & VAN ZWOL, R. (2009). Placing flickr photos on a map. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in Information Retrieval*.

SHEATHER, S. & JONES, C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*.

SLOCUM, T.A., MCMASTER, R.B., C.KESSLER, F. & H.HOWARD, H. (2005). *Thematic cartography and geographic visualization*. Prentice Hall.

SPEARMAN, C. (1987). The proof and measurement of association between two things. *The American journal of psychology*.

SUTTON, C. & MCCALLUM, A. (2006). *Introduction to Conditional Random Fields for relational learning*. MIT Press.

SZALAY, A.S., GRAY, J., FEKETE, G., KUNSZT, P.Z., KUKOL, P. & THAKAR, A. (2005). Indexing the sphere with the hierarchical triangular mesh. Tech. rep., Microsoft.

TEITLER, B.E., LIEBERMAN, M.D., PANOZZO, D., SANKARANARAYANAN, J., SAMET, H. & SPERLING, J. (2008). Newsstand: a new view on news. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information systems*.

TURNEY, P.D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.

VAPNIK, V.N. (1979). *Statistical learning theory*. Wiley.

VINCENTY, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*.

VISHWANATHAN, S.V.N., SCHRAUDOLPH, N.N., SCHMIDT, M.W. & MURPHY, K.P. (2006). Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of the 23rd international conference on Machine learning*.

VITERBI, A. (2006). A personal history of the viterbi algorithm. *Signal Processing Magazine, IEEE*.

WIJNHOVEN, R. & DE WITH, P.H.N. (2010). Fast training of object detection using stochastic gradient descent. In *Proceedings of the IEEE International Conference on Pattern Recognition*.

WING, B. & BALDRIDGE, J. (2011). Simple supervised document geolocation with geodesic grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.