João Mário Gonçalves da Costa

Web Page Classification using Text and Visual Features

September 2014

· U C ·

Universidade de Coimbra

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

A Dissertation
for Graduate Study in MSc Program
Master of Science in Electrical and Computer Engineering

Web Page Classification using Text and Visual Features

João Mário Gonçalves da Costa

Research Developed Under Supervision of
Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves

Jury
Prof. Doutor Fernando Santos Perdigão
Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves
Prof. Doutor Paulo José Monteiro Peixoto

September 2014

Acknowledgments

After six years of good and bad moments, i would like to thank to many people that helped and supported me in this period of time.

I especially want to thank to my parents for all the education and the help that they gave me. For all the effort and sacrifices made to gave me the opportunity to attend university and to become what i am now.

I would like to express my appreciation to Professor Nuno Gonçalves for the help he gave in all the stages of this thesis. For all the opinions and advice so that i could develop this work in the best way.

I would like to acknowledge my immense gratitude to all my friends for the help and the support during this period of time. For all the happy moments and experiences we had together. They turn this six years in one of the best phases of my life.

To all my sincerest thanks and appreciation,

João Costa

Abstract

The world of Internet grows up every day. There are a large number of web pages actives at this moment and more are released every day. It is impossible to perform the web page classification manually. It was already developed several approaches in this area. Most of them only use the text information contained in the web pages, ignoring the visual content of them.

This work shows that the visual content can improve the accuracies of the classifications that only use the text. It was extracted the text features of the web pages using the term frequency inverse document frequency method. As well, it was also extracted two different types of visual features: the low-level features and the local SIFT ones. Since the amount of the SIFT features is extremely high, it was created a dictionary using the "Bag-of-Words" method. After this extraction the features were merged, using all the types of combinations of them. It was also used the Chi-Square method that selects the best features of a vector.

In the classification it was used four different classifiers. It was implemented a multi-label classification, for which we gave unknown web pages to the classifiers, so they could predict the main topic of the web page. It was also implemented a binary classification, for which we used only visual features to verify if a web page was a blog or non-blog.

It was obtained good results that shows that adding the visual content to the text the accuracies improve. The best classification it was obtained using only four different categories, where was achieved 98% of accuracy.

Later it was developed a web application, where the user can find out the main topic of a web page only inserting the web page URL. It can be accessed in "http://scrat.isr.uc.pt/uniprojection /wpc.html".

Keywords: Web page classification, feature extraction, Blogs,term frequency-inverse document frequency, SIFT, low-level.

Resumo

O mundo da internet cresce a cada dia que passa. Existe um enorme numero de páginas web activas neste preciso momento e muitas mais são lançadas a cada dia que passa. É impossivel realizar uma classificação manual destas páginas web. Já foram realizados diversos trabalhos nesta área. A maioria delas apenas utiliza a informação do texto da página web, ignorando o conteúdo visual das mesmas.

Neste trabalho mostramos que o conteúdo visual melhora as precisões dos classificadores que utilizavam apenas texto. Para isso foram extraídas características de texto das páginas web utilizando o método term frequency-inverse document frequency. Foram extraídos dois tipos de características visuais: as características low-level e as características locais SIFT. Sendo que o número de características SIFT é extremamente alto, foi criado um dicionário utilizando o método "Bag-of-Words". Depois de extaídas, foram feitas todas as combinações possíveis entre estes três tipos de características. Foi utilizado também o método Chi-Square que seleciona as melhores características.

Na classificação, foram utilizados quatro classificadores diferentes. Foi realizada uma classificação multi-label, onde introduzindo páginas web desconhecidas pelos classificadores, os mesmos previam o tópico principal dessa página. Foi também realizada uma classificação binária onde apenas foram utilizadas as features visuais para verificarem se uma página web é um blog.

Foram obtidos bons resultados que mostram que realmente adicionando o conteúdo visual ao texto, as precisões dos classificadores melhoram. A melhor classificação foi obtida quando utilizadas apenas quatro categorias diferentes, onde foi obtida uma precisão de 98%.

Posteriormente foi desenvolvida uma aplicação web com o objectivo de um utilizador conseguir descobrir qual o tópico principal de uma página web apenas inserindo o seu URL. Pode ser acedida em "http://scrat.isr.uc.pt/uniprojection/wpc.html".

Keywords: Classificação de páginas web, extração de features, Blogs,term frequency-inverse document frequency, SIFT, low-level.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The world of Internet grows up every day. There are billion and billion of web pages actives at this moment and more are released every day. There are too many information in the Internet that is useful for the population. The number of people that use the Internet is also increasing. With this exponential growth it is increasingly necessary web page classification to simplify and help users of the Internet. Since the number of web pages is not measurable, it is impossible to classify the web pages manually. It is needed an automatic way to perform web page classification. This classification is really important to manage the enormous amount of information that exists in the Internet. The web page classification is also essential to focused crawling (seek for web pages that are relevant to a predefined set of topics), to assist in the development of web directories, helps in the analysis of the topical structure of the web, improves quality of web search, web filtering, assisted web browsing, avoiding spam, focused marketing and more.

A web site, an interconnected set of web pages, is created with well-defined objectives. According to these objectives each of that web pages is idealized by the designers and programmers so as to satisfy in the better way the needs of the target users that the web site seeks. The web pages need to draw attention of the user in a way that he/she enjoys and feels comfortable to use it. Thus, the design of a web page, i.e. the way that it is displayed to the user, gets more and more importance each day. The visual structure of web pages of a particular subject or category tend to have similarities so that users feel familiar with them. Thus, the visual content of the web page contain important information and can help in web page classification. It is in this assumption that this work was realized, using visual content together with the text of the web pages.

People use more and more the Internet as source of information, for diverse tasks (as to buy some product for instance), for entertainment and much more. To search for some web pages that satisfy their needs there are search engines like Yahoo, Google and more. This search engines have algorithms that are constantly navigating in the Internet doing crawl and classifying each web page by their content and other factors. That classification is kept and controlled by an index that have a size over 100 millions of gigabytes (as stated by Google).

1.1 Motivation

The motivation behind this dissertation is based on two works: [6] and in [4]. In these two works, the authors proved that the visual features have important information of the web pages and can be used to classify the web pages in diverse types of classification. Using a binary classification they classified the web pages by two different subjective variables: their aesthetic value (whether one page is ugly or beautiful) and by their recency (whether one page have an old or a new design). Using the multi-label classification they classified the web pages by their topic. They obtained good accuracies, mainly for the binary classification. Our motivation is to improve the state of the art classifications that use only text by integrating them with visual features that, as mentioned before, showed to contain useful information for this task.

1.2 Objectives

Each web page is composed by different types of elements like images, videos, banners, tables, text, publicity and much more that are organized in a way to capture the attention and be able to transmit the information in the best and easiest way possible to the user. The classification of the web pages is performed through the text ignoring the visual component of the web pages. The main objective of this work is to show that the visual content of the web pages can improve the classifications obtained when only the text is used to classify the topic. After extracting the text features and the visual features of the web pages, it was performed systematic tests (using the text, SIFT and low-level features separately and using their combination), in order to understand how the accuracies of the classifiers evolve.

The web page classification problem can be divided in a multi-label classification or in a binary classification problem, depending on the number of categories that it is intended to classify the

web pages. In our work it was implemented one of each type. In the multi-label classification we intend to classify the web pages by their topic: classify the main theme of the information contained in the web page. In the binary classification we aim to find out if one web page is a blog or a non-blog. Knowing that the Blogs are a special type of web pages where it is to hard to perform a classification using their text, since the type of information is constantly changing, it was performed a binary classification where we aim to to find out if one web page is a blog or a non-blog only using their visual features.

1.3 Main contributions

The first contribution is the creation of a data set of seven different categories, that contains the visual content of the web pages and their source code. Each category have 500 web pages. The algorithms was written in C/C++ to filter the text information contained in the HTML files that forms the web page and algorithms to extract the text features of these pages, to be used for the web page classification. To the best of our knowledge, this is the first web page classification by topic using the combination between text features and visual features of the web pages.

It was obtained good results, that show that the visual content of the web pages improve the classification. The best results were obtained using four categories. The best classification was thus 98% (only four, out of 200, web pages were badly classified) obtained using the fusion between the text features and the low level features through the SVM classifier. With seven categories the SVM classifier also had high accuracies. The best accuracy was 92.28% when merged the text features with the low level features. The Naive Bayes had accuracies near 90%. The Decision Tree and the Ada Boost classifier had poor classifications with seven categories. However with four categories their accuracies were always above 70%. In the binary classification that distinguish if one web page was a blog or a non-blog pages the accuracies were all high, with all the classifiers always above 80%.

1.4 Structure of the thesis

This thesis is organized as follows: chapter 2 presents the related works developed in the web page classification area. It explains various different approaches that was done in the past using various type of features, mainly the text features of the web pages. In chapter 3 it is explained

how the extraction of the features of the web pages is done and how the classification process works. Chapter 4 explains how it was created the data set and how it was done the filtering of that data set. Chapter 5 shows all the important results obtained in our work. It also shows a computational study to understand how the classifiers behave with the variation of the number of features. It is discussed the global results and the relation between the results and the time that the classifiers need to be trained and to test. In chapter 6 we show and explain a web application developed to classify web pages by their topic. Finally in chapter 7 we draw some conclusions of this work, presenting the positive and the negative points and we give ideas for future work.

Chapter 2

Related work

In the web page classification many tactics have been developed to classify the web pages. A web page is a computer file usually written in HTML, that contains HTML tags to structure the file, text, hypertext that will navigate to other web pages normally related and anchor texts. Web pages can be classified by their topic (i.e. Arts, Sports, News, Games, etc), by their functional objective (i.e. personal home page, blog, institutional page of an organization, etc), by the emotions that a web page provides to the user, by their aesthetic value and many more.

2.1 Web page classification

To perform the web page classification, many different types of features are extracted from the web page taking into account the approach that will be used. The more common approach used was using the text content of the web page to classify its topic. The first works in this area filtered the text information of the web page. They preprocessed the web pages removing the HTML tags and stop-words, and stemming the words. Later studies reveal that the information contained in the HTML structure could also be useful to help in the web page classification. Recently, also the URL of the web pages is known to contain information that is useful to help in this problem. Other approaches used the information contained in the hypertext using the neighbor pages to help the classification of the web page.

Riboni [15] exploits the information provided by the HTML structure and by the presence of hyperlinks. To exploit the HTML structure he tested five different text sources. They were:

- **BODY**, using the text contained in the BODY tag.
- **META**, using the page descriptions in the META tag.
- **TITLE**, using the page's title.
- **MT**, fusion of the META and TITLE information.
- **BMT**, fusion of the BODY, META and TITLE information.

With these tests he concluded that the information contained in the meta tag and in the title of the web page had more importance to categorize it. To calculate the weight [16] of a term in a document he didn't base only on the frequency of the term but also on the specific tag of the HTML structure (i.e. meta tag). He called this weighting method "Structure-oriented Weighting Technique (SWT)". To use the hyperlink information, Riboni developed a new method where it is not necessary to download the page. It only uses the local information of the hyperlink, more specifically, it uses the anchor of the hyperlink, which is the text presented to the user to identify the link. He also developed a method combining the local and hyperlink representation.

M. Kan and H. Thi [7] approach the problem of web page classification using only the URL of the web pages. This approach is faster than the traditional approaches. It is not needed to analyze full texts of the web page. Their results show that, in some scenarios, this URL method performance approach and sometimes exceeds the performance of the full-text and link-based methods.

S. Tice[21] developed a method of classification of web pages using the Yioop search engine with active learning. Yioop search engine is a search engine similar to Google and Yahoo search engine but only uses a data set of web pages previously entered by the user. In this method he developed a function to the Yioop search engine, to automatically assign "class" words to the web pages. The users can create multiple classifiers, who will be trained first in a small set of labeled documents and then improved after repeated rounds of active learning.

Kwon and Lee [9] created a scheme for Web site classification based on the k-nearest neighbor (k-NN) approach. This approach have three steps:

- **Web page selection**, using connectivity analysis, given a web site (a set of interconnected web pages) the web page selection choose several web pages.
- **Web page classification**, using the k-NN each web page is classified.

- **Web site classification**, the web pages classifications are used to classify the web site.

They used a feature selection method to improve the performance of the k-NN approach and a term weighting scheme to distinguish the importance of terms with different HTML tags.

To avoid the "noisy" information contained in the web pages (i.e. advertisements) , Shen et al. [18] created a new algorithm based on Web summarization techniques to classify web pages. The web page is preprocessed with summarization techniques before the classification. They obtained better results comparing with the classification based on pure-text. Selamat et al. [27] developed a new method to classify the type of news that news web pages contain. This method is based on the principal component analysis (PCA) and class profile-based features (CPBF). Their approach combine two types of feature vectors. One is obtained calculating the weight of all the words using the Term Frequency-Inverse Document Frequency (tf-idf) and using the PCA for feature reduction and selection. To obtain the other vector, they used the CPBF. They select the most regular words from each class and calculate the weight of each word using an entropy method. After the fusion of the vectors, it is used the neural networks for the training phase. Then, the classification is done.

Chen and Hsieh [3] did the classification of the web pages based on a Support Vector Machine (SVM) using a weighted vote scheme. They used two types of features and did two separated train/tests with that features using two SVM models. They used semantic features that are extracted using a latent semantic analysis (LSA) and text features that are extracted using a Web page feature selection (WPFS). After the two SVM models classify the web pages, the results of this classifications are used in a voting scheme to determine to which class it belongs.

Vaghela et al. [22] approach the web page classification problem using Term Frequency (tf). They built a method that calculate the tf of the stem of a term for each HTML tag. The combination between each stemmed term frequency and each tag form a feature.

For the hierarchical classification of the web pages Wibowo and Williams [26] built a method that use a low fixed number of features from pre-categorized training documents. This features are extracted from the beginning of the document. They believe that the first features of the documents have the necessary amount of information to categorize the document.

All this techniques for a web page classification are developed using only the text information on the HTML code of the web page. All of the information provided by the visual content of the web page is ignored. Recent studies used the visual information of the web page to do the

web page classification. They demonstrate that the visual content of the web pages is relevant for their classification and can complement text methods to get better classification accuracies.

N. Gonçalves and A. Videira [6] basing in Boer et al. [4] developed a method of web page classification using visual features. With features extracted from the visual content of a web page, they classified the web pages by their Topic (the category of the web page), by their Aesthetic Value (if a web page is beautiful or ugly) and by their Recency (if a web page have a recent or old layout). They used two different approaches. One of them is creating a descriptor of the web page using their low-level features (color and edge histograms, Gabor and Tamura features). The other approach was using the Bag-Of-Words (BoW) model to transform the SIFT local features into "visual words" and build a dictionary for characterize each web page.

Asirvatham et al.[1] developed a method to do the Web page classification using their structural information. They combined the visual information of the images displayed, the amount of text content and the number and placement of the links in the web page. They built a method to categorize the web pages into three categories: Information pages, Research pages and Personal Home pages.

Kovacevic et al. [8] believed that the visual layout of a web page could be useful in the web page classification. In their technique the web page was represented as a hierarchical structure - Visual Adjacency Multigraph, in which nodes represent the HTML objects while directed edges reflect spatial relations on the browser screen. Using the multigraph information, one is able to define heuristics for recognition of the common sections of the web page.

Based on the image processing techniques and artificial neural networks, M. Mirdehghani and S. Monadjemi [12] developed an automatic system for web page classification by their aesthetic value. They classified the web page into three groups: low quality, moderate and high quality. Since the aesthetic value of an web page depends on the tastes of the persons, to build the train data set they developed an on-line questionnaire.

The usability it is an important characteristic to have in consideration when a web site is evaluated. Therefore, some work have been developed in this area to help web designers. A. Dingli and S. Cassar [5] proposed an Intelligent Usability Evaluation tool that automates the usability evaluation process by employing a Heuristic Evaluation technique using several research-based Artificial Intelligence methods.

2.2 Blogs classification

There are a special type of web pages with an exponential growth in recent times that deserve a special attention. The Weblogs or blogs. With that exponential growth it was necessary to find a way to do the classification of the blogs to help users to locate the topical blogs of their interest. The blog classification is different in several aspects from the other web pages. The content of a Blog is changing constantly, in a daily basis. It is frequently changed by publishing new blog posts. Other problem is the fact that a blog can change the subject from one post to another. It highly depends on the criterion of the blog author. The diversity of topics and the frequent update nature of the blog makes the classification much more harder then the normal web page classification. Due to that fact, some work was been developed to help to solve the blog classification problem. Diverse features can be extracted from the blog properties to use in blog classification, such as the blog posts, the title, the tags, the description and more.

A. Sun et al. [19] studied the effectiveness of tags in blog classification. They compared the classification using the tags of the blogs with the classification done using other type of data (i.e. title, description etc). Their results showed that tags were more effective than features extracted from the blog title and description. Using all the features they got better results.

C. Brooks and N. Montanez [2] analyzed the effectiveness of tags for classifying blog posts. They concluded that tags are useful for grouping articles into broad categories, but they are not so good to indicate the particular content of the post. However, they also concluded that extracting the more important words (using the Term Frequency-Inverse Document Frequency) from the posts can have good results to categorize the articles.

Chapter 3

Feature Extraction

For the web page classification it is used image and text information contained in the web pages themselves. The information contained in the web pages is resumed in feature vectors. In this chapter we will describe how the three types of features are extracted: Low-Level, SIFT and Text features. It will be explained the concept of Bag-of-Words, that is used to build the dictionaries in the SIFT and Text features.

3.1 Visual Features

The concept of feature in computer vision and image processing refers to a piece of information which is relevant and distinctive[23]. For each web page, it can be extracted different types of features vectors. This work was based on [23], where it was used the low-level features and the SIFT features using the Bag-of-Words model. In this subsection we will explain how we extracted this two different visual features.

3.1.1 Low-Level Features

To characterize an image with a vector of features, it can be used many type of low-level features. We built a vector, for each image, with 166 attributes extracting four type of features, that are :

- **Color histogram** - represent the distribution of colors in the image. There are many color spaces and a color histogram can be built in any of them. We based our work on the RGB and HSV color spaces. It was used 32 bins for the color histogram.

- **Edge histogram** - represent the distribution of edges orientations in the image. An edge is a curve that follows a path of rapid change in image intensity. The number of bins was 80.

- **Tamura features** - Tamura et al. [20] propose six types of features : coarseness, contrast, directionality, line-likeness, regularity and roughness. After several tests, they concluded that the first three were the most significant features. In this work we will extract the coarseness, contrast and directionality features. It will be extracted 6 bins of each feature.

- **Gabor features** - it is obtained convolving the image with a gabor filter, that is a linear filter used for edge and texture detection. Gabor features are sensitive to different scales and frequencies. This properties are useful when texture analysis is required. For the gabor features it was extracted 36 bins.

3.1.2 SIFT Features

The scale invariant feature transform (SIFT) is an algorithm developed in 1999 by David Lowe, to detect and describe the local features in images. For any object in some image, SIFT extract the interesting points creating a feature descriptor. Those descriptors can be useful to identify that object in other images. The SIFT features are very popular in local image feature extraction for general images. They are described in greater detail in [11]. There are four main stages to build the set of this features :

- **Scale-space extrema detection** - searches over all scales and image location

- **Keypoint localization** - for each keypoint the interpolated location of the extremum is calculated, to improve matching and stability.

- **Orientation assignment** - for each keypoint, one or more orientations are assigned, based on local image gradient directions.

- **Keypoint descriptor** - for each keypoint, a descriptor vector is computed. This results in a feature vector containing 128 elements.

3.2 Text Features

Text mining is the process of selecting important information from a text. That is, text mining search for standards in a raw text. There are many text mining methods to evaluate the importance of a word in some document. One important method is the Term Frequency-Inverse Document Frequency (tf-idf)[14]. With this framework we can transform the text information into a Vector Space Model (VSM) [17].

Vector Space Model is an algebraic model to represent text information as a vector. In other words, it is a space where the text is represented as a vector of values instead of a text string. Each value of the vector represents a feature extracted from a text document. This features are related to a single word and measure the importance of that word in the document. The values of this vector could be not only the importance (td-idf) but also the presence of that word in the document.

The importance of a single word in a document text can be calculated using the tf-idf, as we mention before. In a simple way, the tf-idf is calculated using the number of times that a word appears in a single document multiplied by an inverse proportion of that word in all the documents.

The procedure to calculate this value have some small changes depending on the type of applications that will use it for. In a generic way, in a set of documents D, a word w, and one single document $d \in D$:

$$w_d = f_{w,d} * \log(\frac{|D|}{|D : winD|}) \tag{3.1}$$

where $f_{w,d}$ is the number of times (absolute frequency) that the word w appears in the single document d , $|D|$ is the size of the set of documents, and $|D : winD|$ is the number of documents where the word w appears in document D [16]. The existence of a word repeatedly in a small section of the all documents, have a high value of tf-idf. It is in this type of word we are interested in. It is important to differentiate the groups of documents that we have. The existence of a word in the majority of the documents, like articles and prepositions, will produce a small value of tf-idf. That type of words are called "stop words", they are not important to categorize or differentiate one document from another, since they appear in almost all the documents. We will explain in much detail the concept of categorizing and grouping documents in chapter 4. Giving a set of words w_i, that can be built using the bag of words model, it can be easily calculated the value of the tf-idf for each word w_i in a document. For this, it is only needed to analyze the

document and running a sum of $f_{w,d}$ and analyze the full set of documents and running a sum of $f_{w,D}$. With this values of w_i it is created one VSM, that now represents one single document.

Sometimes, some documents can have a spam of certain words, that will have a high value of tf-idf. This don't mean that this word have more importance than others. In longer documents we can have the same problem. One word can appear with more frequency in longer documents than in shorter ones. To avoid this problem we need to normalize our vectors ($\hat{v} = \frac{\vec{v}}{||\vec{v}||_p}$ (for which we used the Euclidean norm – $p = 2$)).

To extract the features of each document, it is needed to make the calculation of the tf-idf value of each word that is in the dictionary. The algorithm 1 explains how to create the IDF value of each word. After this, the algorithm 2 extracts the tf value of each word and build the features vector of the document.

Algorithm 1 Calculation of Inverse Document Frequency

Input: Document with the Dictionary of words

Input: Text Documents to extract features **Output:** Document with words and their respective IDF value

1: For each word

sum the number of documents where it appears.

2: For each word

calculate the logarithm value of the division between the total number of documents used and the number of documents where the word appears (idf).

Algorithm 2 Calculation of Term Frequency and build features vector

Input: Text Document to extract features

Output: Vector of features

1: For each word

sum the number of times the word is in the document (tf).

2: For each word

multiply the tf and the idf value.

3.3 Features fusion

One of the objectives of this work, like it was explained in section 1.2, it was to show that visual features have important information to help in the classification of web pages. To accomplish this, we need to consider the fusion of features of different types. All the features extracted are already normalized to the [0,1] interval. We then created several feature vectors, using all possible combinations between the three types of feature vectors.

Since the difference between the number of text features and the number of visual features is extremely high (only 666 visual features and more than 13000 text features), it was used the Chi Square method [10] to select the most important features to keep to the training phase. The Chi Square method evaluate the importance, or the worth, of a feature to the class that it belongs by computing the chi-squared statistic value (basically, by measuring the difference between expected and observed frequencies).

3.4 Bag of Words

The concept of bag-of-words is widely used in classification problems. Due to the huge number of features (more than 200000), visual and text, the extraction of features and the train of the classifiers would be too heavy computationally. Many features wouldn't contain any type of relevant value to the classification. Thus, it is used the concept of bag-of-words. As the name implies, it is created a "bag" with the words that have more importance. Based on this "bag" (dictionary), the features are extracted.

- **Text Features** The dictionary of text words can be built in diverse ways, like the most repeated words of each category, the most repeated words in all the documents, using a threshold for the idf values. After the dictionary creation (the "bag"), the features are extracted from the documents. After that, the document is in the VSM. That vector can represent the absolute frequency of the words (number of times), can represent the presence or absence of the words (binary) and can represent the importance of the word in the document or sentence that is being evaluated (tf-idf). In this work it is used the importance of the words in the document, as described in section 3.2. For each word in the dictionary, it is calculated its tf-idf value.

- **Visual Features** As in the text features, the bag-of-words for visual features have the same concept. But in this situation, they are not text words in the image. It is needed to create visual words. It is firstly detected the keypoints of all the images, using the functions of openCV library, and then the descriptors of the corresponding keypoints are extracted. After this the descriptors are quantized into visual words, using the *BOWKMeansTrainer* of the openCV library, to create the visual dictionary. Having the dictionary formed, the SIFT features extraction is now made like in the text features. It is searched for every visual word of the dictionary, in all the images.

3.5 Machine Learning and the Classifiers

Machine learning is a sub-field of Artificial Intelligence (AI) devoted to the development of algorithms and techniques that allow the machine to learn to recognize patterns, to predict future trends or behavior, and more, following well stipulated rules by humans. As it was explained before, the Internet is not measurable and this makes it impossible to perform the web page classification manually one by one. Due to this, the machine learning is the best option to realize the web page classification. Virtual assistants, search engines, practical speech recognition, online recommendations, fraud detection and spam filters are some examples that use machine learning.

There are too many ways to realize machine learning. There are supervising learning, where the computer is exposed to a set of examples that are previously labeled, to establish a set of characteristics or rules enabling it to later identify and classify new examples. Anther approach of machine learning is the unsupervised learning. In this type of machine learning it is not given any type of indication to the computer about some type of grouping. All the examples are given to the classifiers, without any type of label. The classifier find out common characteristics to group different types of data.

Machine learning algorithms, when conducting the training, analyze the features and adjust their parameters (weights, thresholds and more) to maximize their performance. This process of parameter adjustment is the motive of the term “learning”.

In our work we used supervised learning. It was used four different classifiers provided by the opencv library. They are :

- **Naive Bayes**: it is calculated the probability of each document belong to one category. This probability is calculated using the following equation:

$$p(c_j|d) = \frac{p(d|c_j)p(c_j)}{p(d)}$$

where $p(c_j|d)$ is the probability of instance d being in class c_j, $p(d|c_j)$ probability of the generating instance d given class c_j, $p(c_j)$ = probability of occurrence of class c_j and $p(d)$ = probability of instance d occurring. The number of features is huge and makes this probability hard to calculate. For that, it is assumed that each feature is independent from the others.

- **Support Vector Machine**: represent each example of each category as one point in the feature space. Then SVM divides that feature space in a well-defined way to group the points of each category in the same side of the division. When a new example is introduced, it is mapped as a point in the same feature space and it is predicted to belong to a category based on which side of the divided space the point is.

- **Decision Tree**: as the name implies, it is created a tree where the leaves are the labels and the branches (paths to reach the labels) are built by the features rules. This classifier is created organizing a series of questions and conditions in a tree structure. When a test record is introduced in the classifier, it performs a series of questions about the attributes of the test record. Each time the classifier receive an answer, a follow-up question is asked until the classifier reach a conclusion about the label that the test record belongs.

- **Ada Boost**: It can be used to improve the performance of other learning algorithms. Ada Boost calls several times the other learning algorithms (weak classifiers). The output of the weak classifiers is combined into a weighted sum that represents the final output of the boosted classifier. It was used the decision tree as weak classifier.

3.6 Chi-Square - Feature selection method

Feature selection, in machine learning, is the process of selecting the most relevant features from a set of features. Relevant features are the features that best characterize each category or class. Selecting the relevant features and eliminating the irrelevant ones, can lead to the improvement of the classifiers accuracies. Since the number of features is reduced when the feature selection

is made, the computational time of the training process is also reduced. Reducing the number of features also helps to reduce the overfitting. Feature selection is also important for the data analysis process, to better understand which features are more important for each class.

In our work, it was used the Chi-Square criterion (χ^2) [10] to accomplish the feature selection. This criterion evaluates if a feature is relevant by making the calculation of the chi-square statistic value with respect to the class. The chi-square statistic value of a feature is calculated by the following formula :

$$\chi^2 = \sum_{i=1}^{rows} \sum_{j=1}^{cols} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} is the observed frequency and E_{ij} is the expected (theoretical) frequency, asserted by the null hypothesis. If the chi-square value is high, means that the sample have a lower probability to be according the expected frequencies and need to be rejected. This criterion of feature selection is represented in algorithm 3.

Algorithm 3 Feature Selection using Chi-Square Criterion

Input: Data Matrix (M×N) ▷ M represents the number of samples, and N the number of features

Input: Number of classes C.

Output: Top K features

1: For each feature and class

Find the mean value corresponding to each feature.

2: Then for each feature

Compute the mean value of the classes mean values.

3: For each feature

Compute the Expected and Observed Frequencies for all features

- Expected Frequency is equal to the size of samples in each class.

- Observed Frequency is the number of frequencies obtained for each sample of each class.

4: For each feature

Compute the chi-squared value for each feature.

$\chi^2 = \frac{(ObservedFreq - ExpectedFreq)^2}{ExpectedFreq}$;

5: Sort the chi-squared values and choose the M features with the smallest sum of all values.

Chapter 4

Web Page Classification

As mentioned previously, there are several ways to perform the web page classification. Web pages can be classified by their topic (i.e. Arts, Sports, News, Games, etc), by their functional objective (i.e. personal home page, blog, institutional page of an organization, etc) by the emotions that a web page provides to the user, by their aesthetic value and many more. In N. Gonçalves and A. Videira work [6], they classified the web pages by their Aesthetic Value, by their Recency and by their Topic, using low-level and SIFT features. In the web page classification by their Aesthetic Value, the web pages are classified as "beautiful" or "ugly". The concept of a "ugly" or "beautiful" web page change from person to person. They obtained the data set for training the classifiers consulting some websites and some articles [23]. The web page classification by their Recency is also a binary classification. The web pages are classified by the temporal value as "new fashioned" or "old fashioned". In the web page classification by their topic, the web pages were classified according to their main subject. They obtained good results.

In this work, it were performed two types of classifications using visual and text features. One was to classify a web page by topic (multi-label classification) and another was if a web page is, or not, a blog (binary classification). To create the data set, it was developed a script to render the web page and to obtain the HTML code of each page automatically. To render the web page it was used a command-line utility named CutyCapt. To obtain the HTML code it was used the command-line tool cURL. All the binaries were implemented in Linux.

4.1 Web page Topic

In order to test and to train the classifiers, the existence of a data set is vital to this work. All the data sets available that we found were not suitable to our work due to the lack of calibration. Thus, we created a data set based on the Alexa popularity rankings [24]. All the websites were extracted from this site. Alexa is a web service site that shows the most visited sites for categories (it is a well-known ranking site).

This classification was started with fourteen classes. They were Arts, Business, Computers, Games, Health, Home, Kids and Teens, News, Recreation, Reference, Science, Shopping, Society and Sports. Back then, we had, for each category, 1000 web pages, where 10% of them were used to test. After several tests with visual features, it was concluded that this automatic data set based on Alexa popularity rankings was not the best. There were several web pages that was in the wrong class. All the classes were examined, and it was necessary to filter the web pages that were not in the most adequate class. The figure 4.1 shows some bad classifications from Alexa site.

After manual filtering, it was reduced the number of classes to seven. It was also reduced the number of web pages in each class to 500. The classes removed were too ambiguous. It was performed the same tests with the filtered classes and the accuracies improved. The final seven categories are Arts, Games, Health, News, Science, Shopping and Sports. In the classification of web pages, each topic has some visual and text characteristics that allows us to distinguish between them. It is important to mention that all the web pages used were written in English. The figures 4.2 and 4.3 show some examples of web pages of each class. It is possible to distinguish each class by visual inspection. On the manual filtering, we observed some specific characteristics of each category. In the Arts web pages for instance, it can be seen big pictures of persons (artists mostly) and draws followed by small descriptors. In this category words like "*movies*", "*music*" and "*draws*" appears with much frequency. The News class is characterized by a lot of text followed with images. In Health appears several pictures of Doctors, using white cloth. Words like "*health*", "*care*", "*diseases*" are frequents in this type of web pages. The Games web pages have videos and big banners with vivid colors. "*Gaming*" and "*play*" are typical words from games web pages. In the Science web pages, pictures of animals and sights are common. Words like "*biology*", "*weather*" and "*earth*" can be found with frequency. Shopping class images with little descriptions with the price of the objects are typical. "*Buy*", "*Sell*" are common words.

Figure 4.1: In the left figure a News web page was classified by Alexa as Arts. In the right figure a Science web page that was classified by Alexa as Health.

The Sports class can be distinguish from the others by the sports symbols and objects like balls. Big pictures of players, stadiums and pavilions full of people. Words like "*football*", "*basketball*" and "*clubs*" are some examples of this class. However, none of this characteristics was obtained performing tests.

4.1.1 Preprocessing HTML files

As stated, to analyze the text of the web pages it was used documents, in HTML format, that contained the source code of each web page. This HTML documents are semi-structured files that contain tags that are used to structure and organize the information that it will be displayed through the web browser.

In our work it was used the text contained in the web pages. It was necessary to filter that information. All the HTML tags, URL, numbers, punctuation, and more were removed. To remove the HTML tags, it was used ELinks that is an advanced and well-established feature-rich

Figure 4.2: Examples of web pages from the classes.

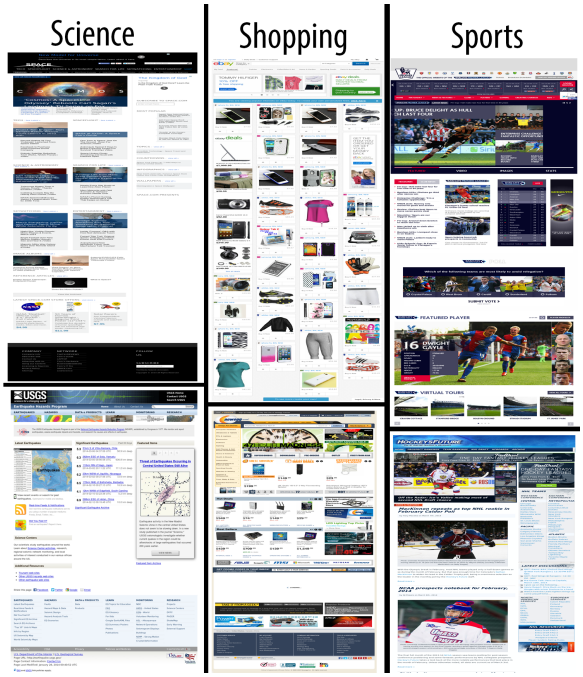

Figure 4.3: Examples of web pages from the classes.

text mode web browser[25].

It was created the dictionary that it would be used. This dictionary contained all the words

(without repetition) that existed in all of the files that would be used to train the classifiers. However, that dictionary had more than one hundred and fifty thousand different words. As it will be shown in the subsection 5.3, the increase of the number of features, also increase the time to train of the classifiers. Thus, it would be impossible to use all of the web page words.

There were many words that appeared only once, many of which could be "trash" created by the filter referred previously. It was filtered the words of the file that contained this dictionary by their inverse document frequency (idf), as explained in chapter 3. Since each class has 450 documents to train, it seemed reasonable to not include words that were present in less than 10 documents. Also, it was removed words that were present in all the documents. After this filtering the dictionary was reduced to 13720 words.

4.2 Blog

A blog is informational or a discussion web page where the information is introduced in a particular form. This particular form is called "posts". Usually blogs are too long, containing many posts. These are typically displayed in reverse chronological order. This kind of web page can have many functions, such as: personal diary or to discuss a particular subject. There are also many types of blogs, depending on the type of posts. The posts of a typically blog have text and images. However, there are blogs where the posts are only text, photographs (photoblogs), videos or music. The author of the blog can give the readers the ability to leave comments about the posts and discuss/interact with other readers.

Nowadays there are a new type of blogs, they are called "multi-author blogs". As the name implies, this type of blogs have many authors. These blogs are professionally edited and belong to newspapers, universities and another similar institutions.

To construct the data set of blogs, it was used the Google search engine.

In this binary classification it is done the distinction between a general web pages and blogs, using visual features. Since the text of the web pages was not used, the web pages don't need to be in the same language. It were used 800 blogs and 800 non-blogs (selected randomly from the 3500 web pages used for the topic classification), using 10% of them to test.

The figure 4.4 shows an example of a blog web page.

Since it was only used the visual features, non-blog web pages that are long and have the main

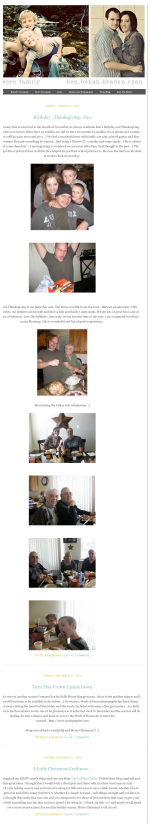

Figure 4.4: Example of a blog web page.

information concentrated in their center are similar to the blog web pages. It can be confused at the human eye when they are not analyzed carefully. In figure 4.5, it can be seen an example of a non-blog web page that were classified as Blog.

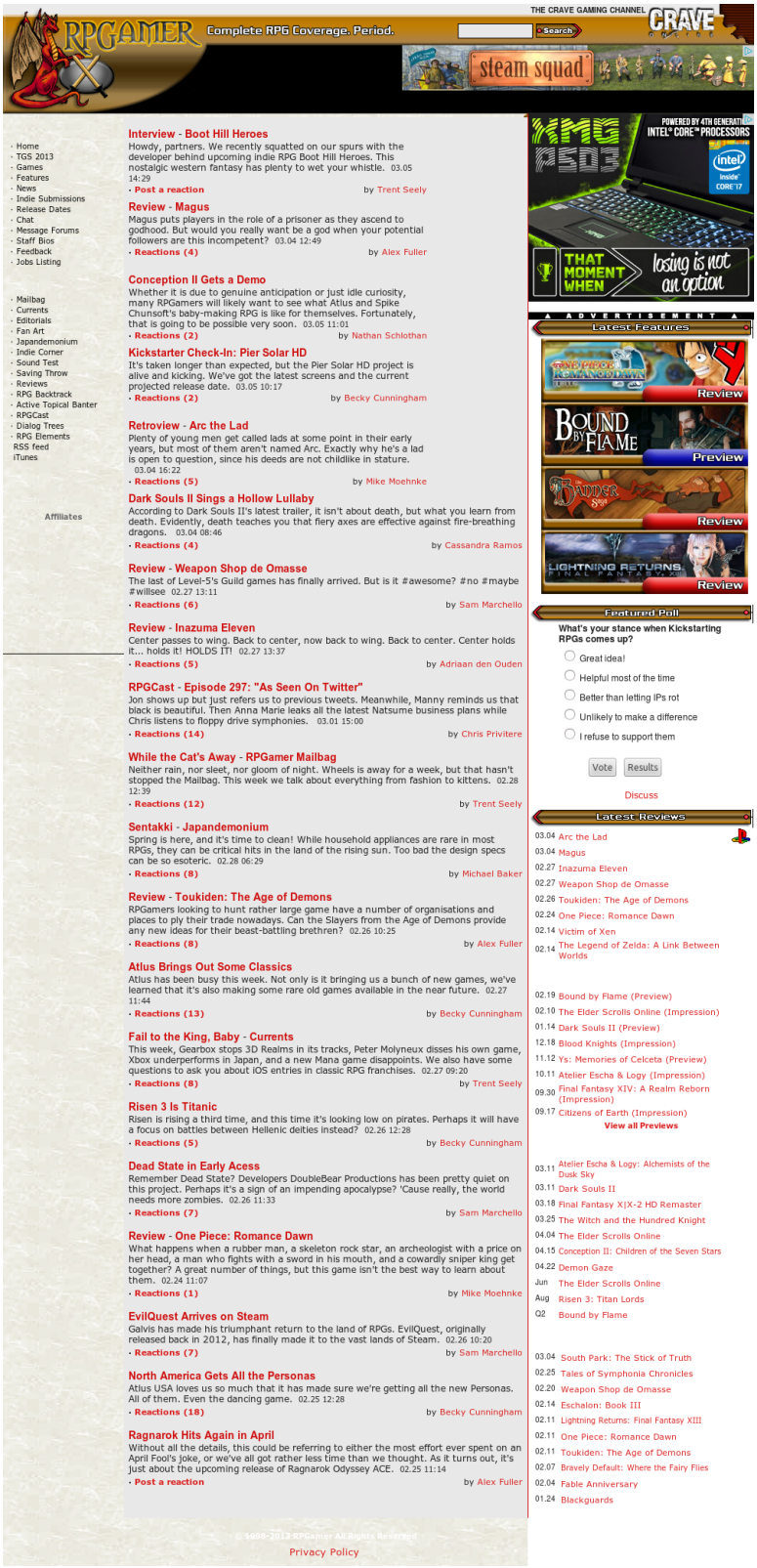

Figure 4.5: Example of web page that was classified as Blog, due to its visual similarity with many blogs.

Chapter 5

Results

5.1 Web Page Topic Results

In this section it will be presented a summary of the classification results of the web pages by topic. It was conducted two different experiments. The first one using seven classes, cited above in chapter 4. The training was performed using 450 web pages of each class. The tests was done with 50 web pages of each category which were not used in the training process.

For the second experiment it was chosen the four "best" classes (the classes that presented best predictions in the first experiment), which are : Games, Health, News and Shopping. The training and the tests were performed with the same amount of web pages that was used in the first experiment.

5.1.1 Experiment 1 - seven classes

Low-Level and SIFT Features

The best result with the low-level features in this experiment was 38.45% of accuracy with Decision Tree classifier using 25% of the features selected using the Chi-square method. This result is not much better than others with others classifiers. For the Naive Bayes classifier the results were 31.14% with 25% of the features. In the SVM, we obtain the worst results, with 22% of accuracy using all the features. With 25% of the features, the Ada Boost had 36% of accuracy. As Gonçalves and Videira [6] concluded in their work, for this type of classification the low-level features, when used alone, present poor results. It is needed more complex features to

improve the results. However, this doesn't mean that they are not important. As we will see in the next results they contain useful information to help the classification of the topic of the web page, when mixed with other features.

With SIFT features using the Bag-of-Words model we got better accuracy when comparing with low-level features in two classifiers. With the Naive Bayes, it was obtained 51.71% of accuracy, and, with the SVM, 45.71% of the classifications were correct. In the Ada Boost and the Decision Tree classifiers it was obtained worse results, with 33.14% and 34.85% of accuracy, respectively.

Merging the two vectors that was explained above and using the Chi-Square method to select the best it was obtained better accuracies in some classifiers. The Naive Bayes classifier obtained 52.57% of accuracy using all the features. The worst accuracy was obtained with the Ada Boost classifier that had 32% of right preditions using 25% of the features. The Decision Tree had their best accuracy when used only 25% of the features with 40% of right preditions. The SVM was a better with 40.28% using 50% of the features.

Text Features

Using only the text features it was obtained high accuracies in some of the classifiers. As figure 5.1 shows the best accuracy was 92%, performed by the SVM classifier. It was obtained using the SVM classifier with 50% of the features selected using the Chi-square method. Due to the high number of features, with our resources the maximum number of features to perform the tests with the Naive Bayes classifier was with 2000 features. With this classifier it was obtained 85.15% of accuracy with the maximum number of features. Maybe it would be possible to improve this accuracy with more features. With the Decision Tree and Ada Boost classifiers it was obtained worse results. The best accuracies using these two classifiers was 55.71% and 49.42% respectively.

Text Features mixed with Low-Level Features

The results of the classifiers using the text features had a small improvement when were added the low-level features. With the SVM classifier it only improved 0.28%. It was expected that this classifier wouldn't improve much more since the accuracy was already too high. In the Decision Tree classifier the accuracy remained the same. The Ada Boost classifier got the biggest improvement performing a 54% of accuracy. In the Naive Bayes classifier, with the maximum

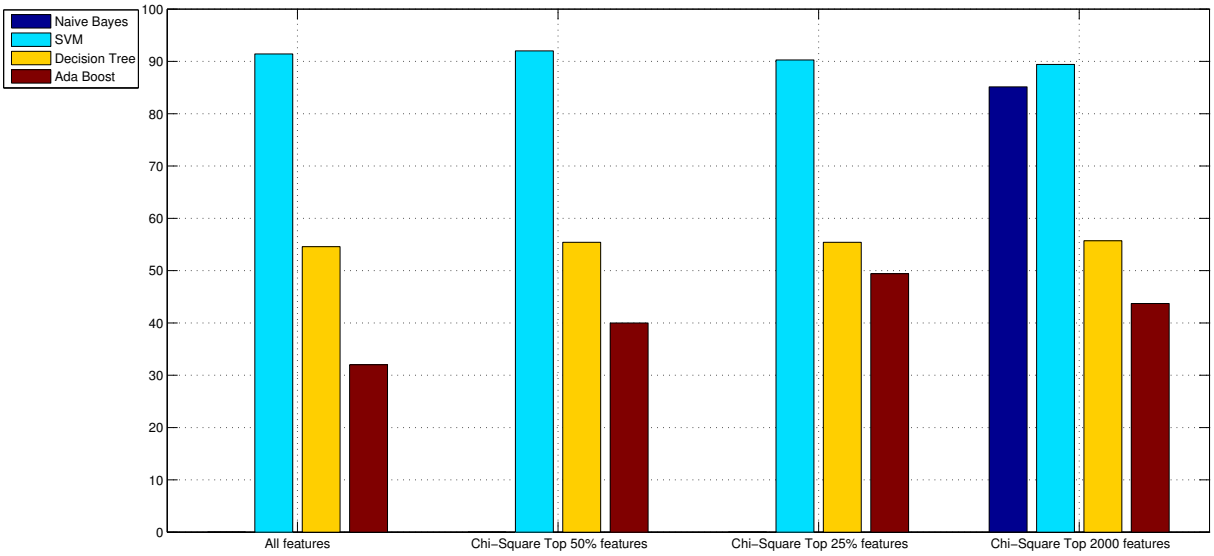

Figure 5.1: Best prediction results for the Topic web page with **seven categories** for four different classifiers, using the **text features**.

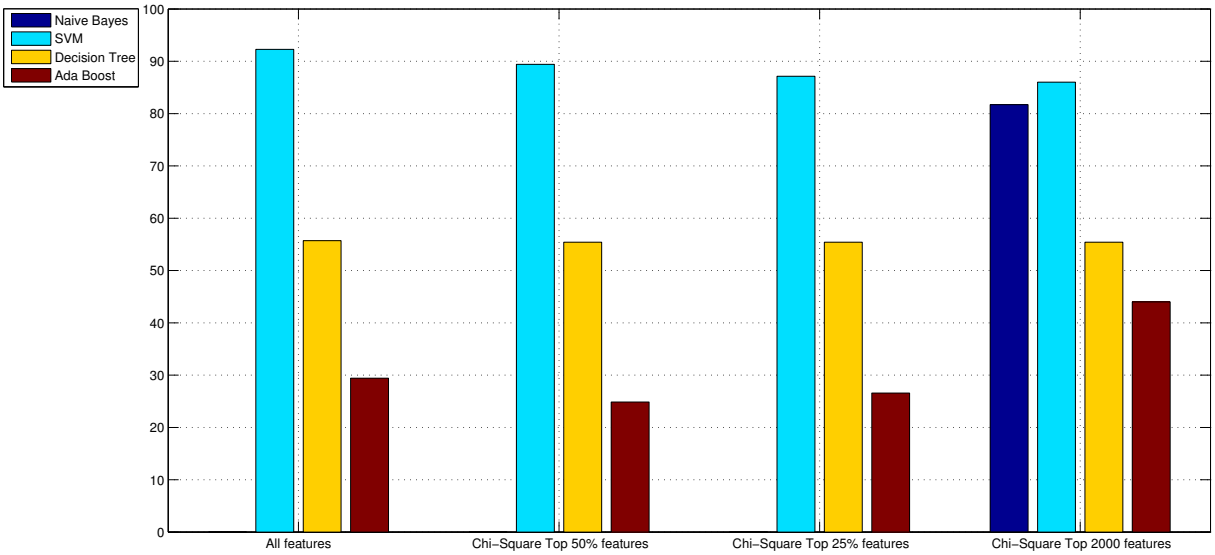

Figure 5.2: Best prediction results for the Topic web page with **seven categories** for four different classifiers, using **text features** merged with **low-level features**.

of 2000 features, it was obtained 81.71% of right predictions. In figure 5.2 it can be seen the accuracies of all the classifiers with a different number of features selected by the Chi-Square method.

Text Features mixed with SIFT features

Adding the SIFT features of the web pages to the text features, the Ada Boost classifier and Decision Tree had improvements. The best accuracy with the SVM classifier was 91.71% with the entire vector of features. Despite being a high accuracy, it was not better than when used the text features isolated. In the Decision Tree classifier it was obtained 59.42% of right predictions and with the Ada Boost classifier they were 55.42% . With Naive Bayes it was obtained 77.71% of accuracy. In figure 5.3 it can be seen the accuracies of all the classifiers with different number of features selected with the Chi-Square method.

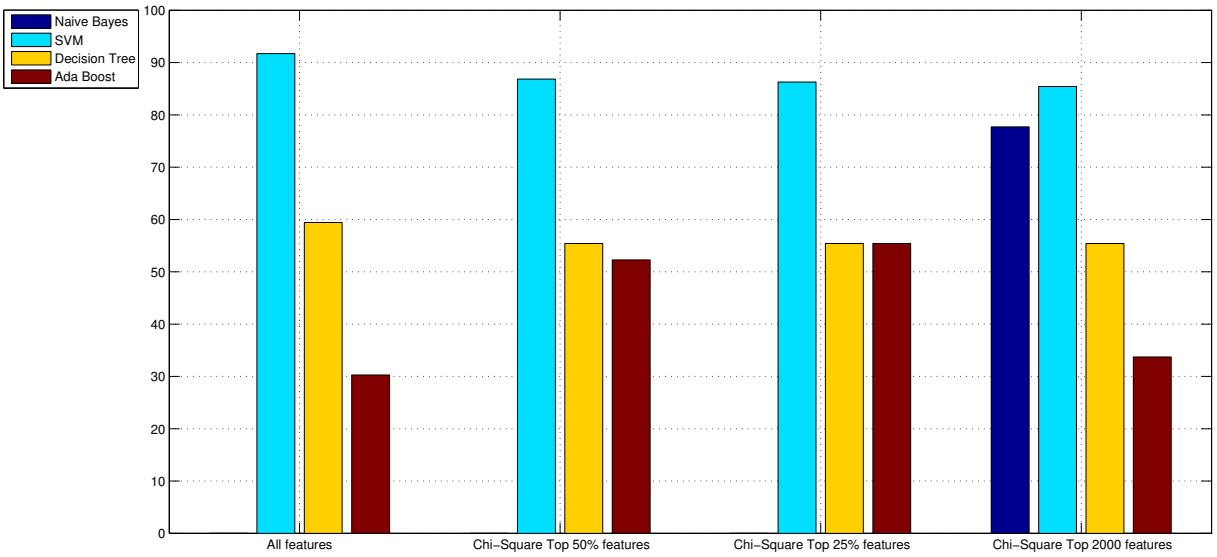

Figure 5.3: Best prediction results for the Topic web page with **seven categories** for four different classifiers, using **text features** merged with **SIFT features**.

All Type of Features mixed (SIFT + Low-Level + Text)

Contrary to what was expected, when mixed all the type of features the accuracies were not better, compared with the classifications using only the text features. The results were lower. The best accuracy was obtained using the SVM classifier, with 90.85% of right predictions. The Naive Bayes and Ada Bost classifiers had 76% and 50% of accuracy with 2000 features. With all the features, 55.71% was the best accuracy using the Decision Tree classifier. In figure 5.4 it can be seen the accuracies of all the classifiers with different number of features selected with

the Chi-Square method.

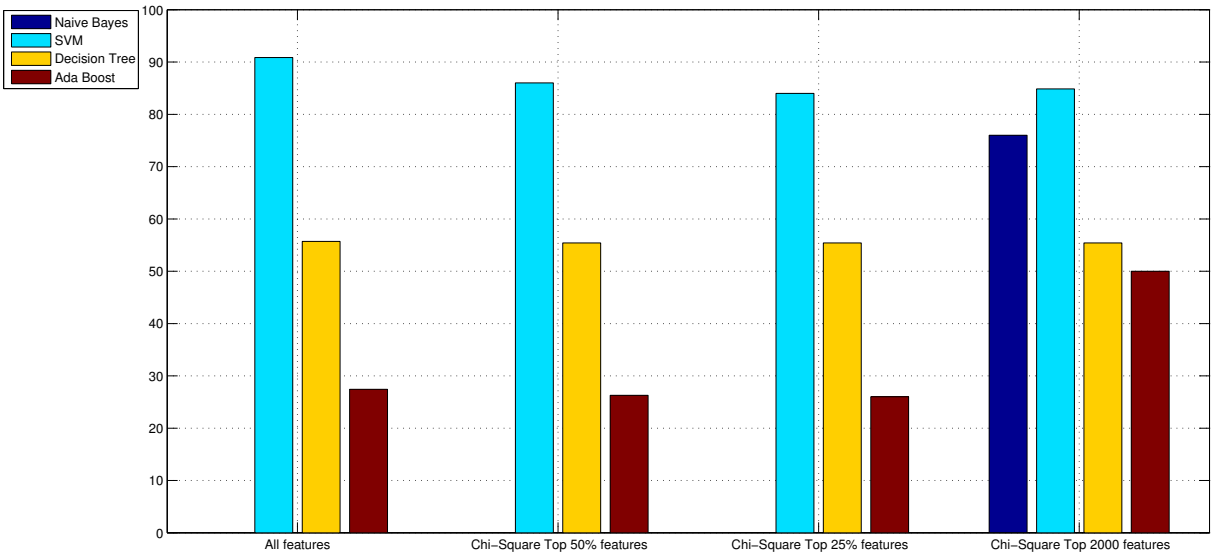

Figure 5.4: Best prediction results for the Topic web page with **seven categories** for four different classifiers, using **all type of features**.

5.1.2 Experiment 2 - four classes

Low-Level and SIFT Features

Using only the low-level features, it was obtained good results, when compared with the same tests with seven categories. With the Naive Bayes and SVM classifiers the accuracies obtained were poor. With 41% and 25.50% respectively. The best result was obtained with the Ada Boost classifier, where 87.50% of predictions were correct. With the Decision Tree classifier it was obtained 62.50% of accuracy.

With SIFT features using the Bag-of-Words model we got better accuracy when comparing with low-level features in the Naive Bayes and in SVM classifiers. With the Naive Bayes, it was obtained 70% of accuracy. With the SVM, 72.50% of the classifications were correct. In the Decision Tree classifier the accuracy was the same. With Ada Boost was better, with 51% of accuracy.

With the two types of features mixed the results were worse when compared with the results only using the SIFT features. The best result combining this two types of features is 70.5% of

accuracy, that was obtained with the Naive Bayes classifier, using the entire number of features. With Decision Tree it was obtained 60.50% using 50% of the features selected by Chi-Square method and with Ada Boost classifier it was 58.50% using 100% of the number of features. The SVM had 64% of accuracy.

Text Features

Using only text features it was obtained remarkable results using all the classifiers. As figure 5.5 shows the best accuracy was performed by the SVM classifier with 97.50% using the entire vector of features. With Naive Bayes it was obtained 90% with the top 2000 features selected using the Chi-Square method. Contrary to what had happened in the tests with seven categories, Decision Tree and Ada Boost classifiers had high accuracies. The Decision Tree using all the features got 87% of right predictions. With 50% of the top features selected by the Chi-Square method, Ada Boost got 89% of accuracy.

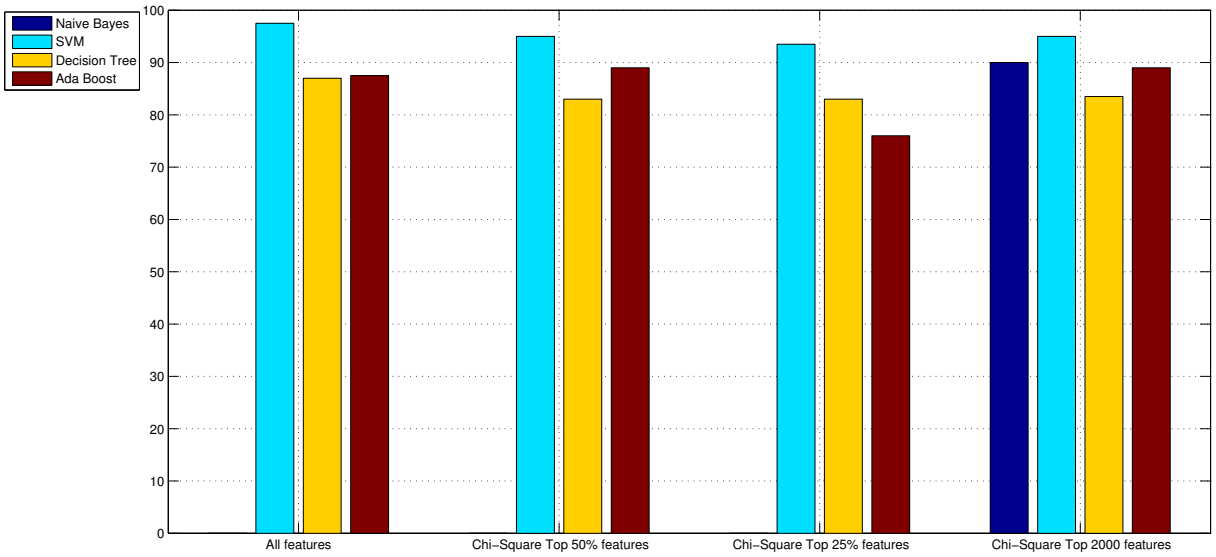

Figure 5.5: Best prediction results for the Topic web page with **four categories** for four different classifiers, using **text features**.

Text Features mixed with Low-Level Features

The best accuracy of all the tests was achieved here. With the SVM classifier, using the entire vector of features, it was obtained 98% of right predictions. The accuracy using Ada Boost

classifier improved 5.5%, when compared when using only text features. A high accuracy of 94.50%. The Naive Bayes classifier obtained 89.5% of right predictions using 2000 features. The Decision Tree classifier also had a high accuracy of 87%. In figure 5.6 it can be seen the accuracies of all the classifiers with different number of features selected by the Chi-Square method.

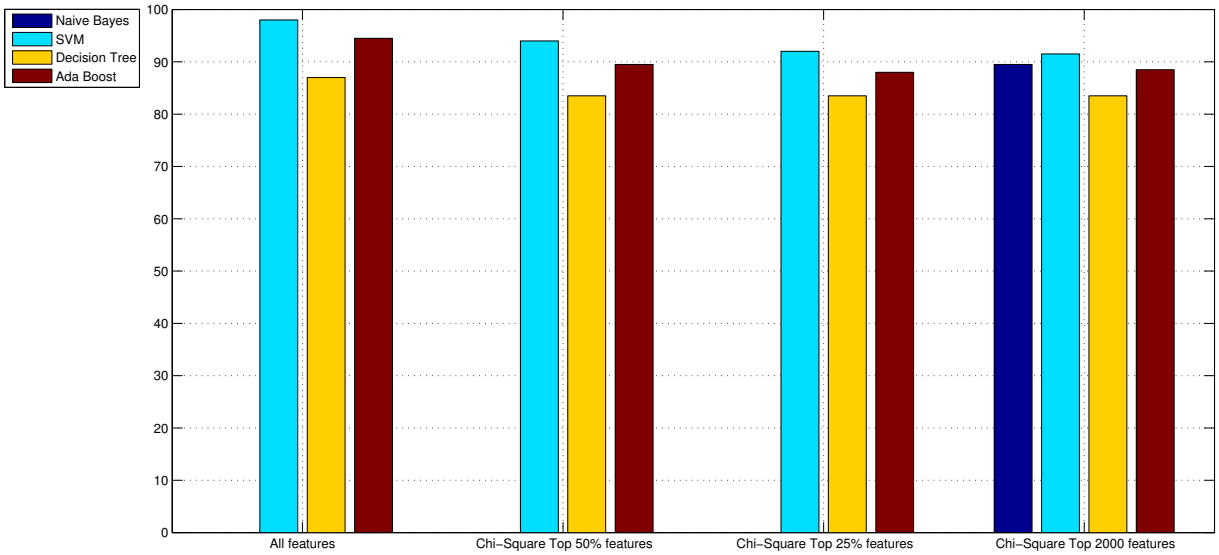

Figure 5.6: Best prediction results for the Topic web page with **four categories** for four different classifiers, using **text features** merged with **low-level features**.

Text Features mixed with SIFT Features

Adding the SIFT features to text features improved the results with the Ada Boost classifier, using the entire vector of features. A 90% of accuracy was achieved. With the Decision Tree classifier the accuracy was the same than that using only the text features. The best accuracy was obtained again with the SVM classifier with 97% of accuracy with 100% of features. The Naive Bayes had 88.5% accuracy. In figure 5.7 it can be seen the accuracies of all the classifiers with different number of features selected by the Chi-Square method.

All Type of Features mixed (SIFT + Low-Level + Text)

The results using all the different type of features mixed were also high. The SVM with an accuracy of 96.50% had the best classification. Decision Tree classifier had the lowest accuracy with 87%. The Naive Bayes had 90.5% accuracy. Ada Boost had a high accuracy of 94%. In

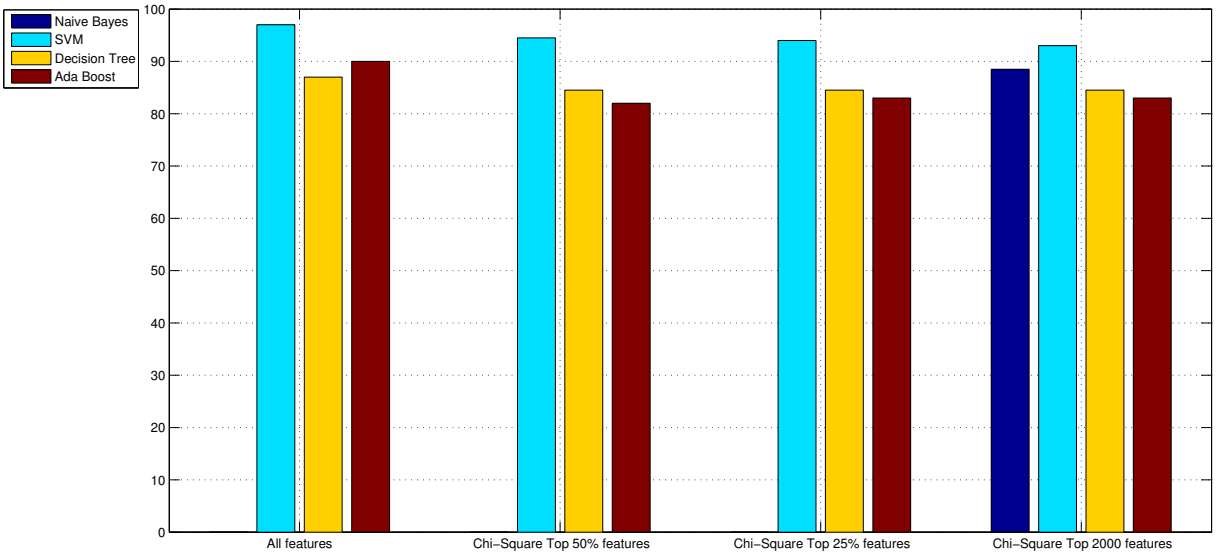

Figure 5.7: Best prediction results for the Topic web page with **four categories** for four different classifiers, using **text features** merged with **SIFT features**.

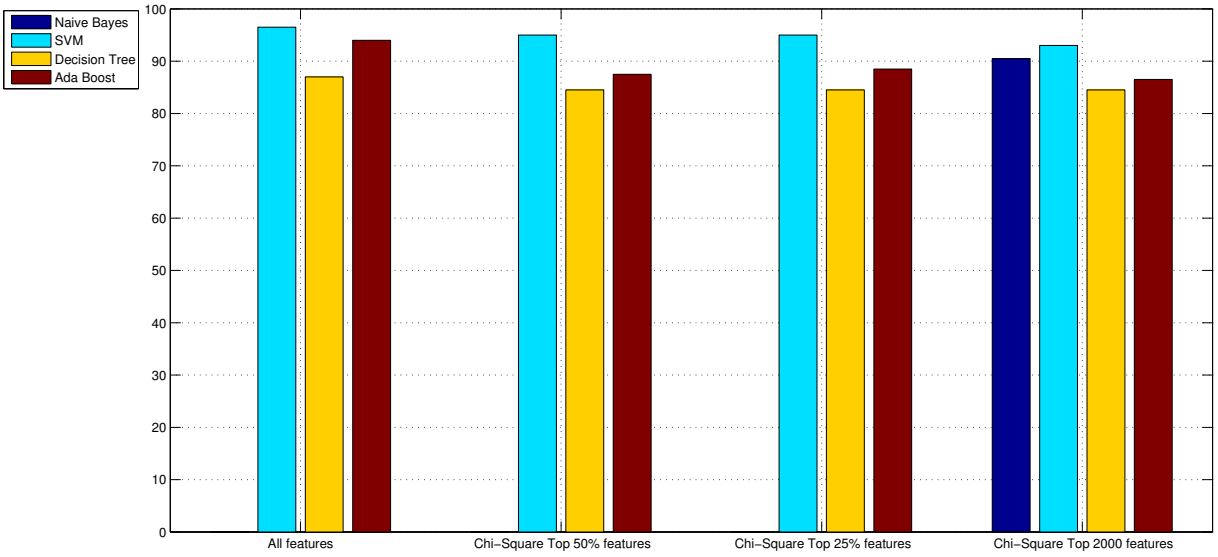

Figure 5.8: Best prediction results for the Topic web page with **four categories** for four different classifiers, using **all the type of features**.

figure 5.8 it can be seen the accuracies of all the classifiers with different number of features selected with the Chi-Square method.

5.2 Blogs Results

In blogs it was done a binary classification. It was to distinguish the web pages that are blogs and the ones that aren't, using only visual features. Two experiments were done. The first one using only 400 web pages (360 of each class for training and 40 of each class for testing) that was blogs and 400 that weren't. In the second one 800 (720 for training and 80 for testing for each class) web pages were used for each class.

As the figure 5.9 shows the best accuracy in the first experiment was obtained using the Decision Tree classifier with 85% of right predictions using the SIFT features. The same accuracy was obtained using the low-level features. The Naive Bayes and the SVM classifiers had a poor performance getting 63.75% and 56.25% of accuracy, respectively, using the low-level features. Using the SIFT descriptor of the web pages in this two classifiers it was obtained good results, with the SVM obtaining 83.75% and the Naive Bayes with 78.75%. With the Ada Boost classifier it was obtained 78.75% with the low features and 72.5% with the SIFT features.

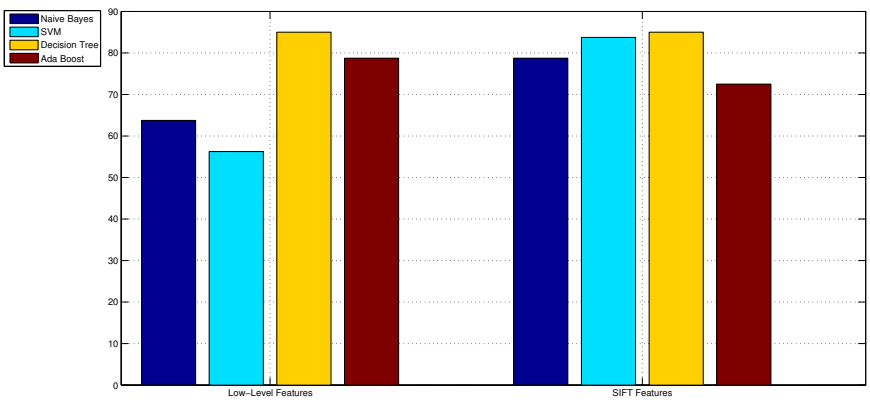

Figure 5.9: Best prediction results for the Blog classification using **360 images** for training and **40 for testing** of each category with four different classifiers, using SIFT and Low-Level features.

In the second experiment (increasing the number of web pages to the double) the accuracies increased slightly in all the classifiers but with the Decision Tree, with the low-level features. As it can be seen in the figure 5.10 the best result was obtained using the Ada Boost classifier with 82.50% of precision using low-level features. With the SIFT features the results decreased with the Naive Bayes and with Decision Tree. The SVM had the best accuracy of 84.38%. In figure 5.10 it can be seen all the classifications.

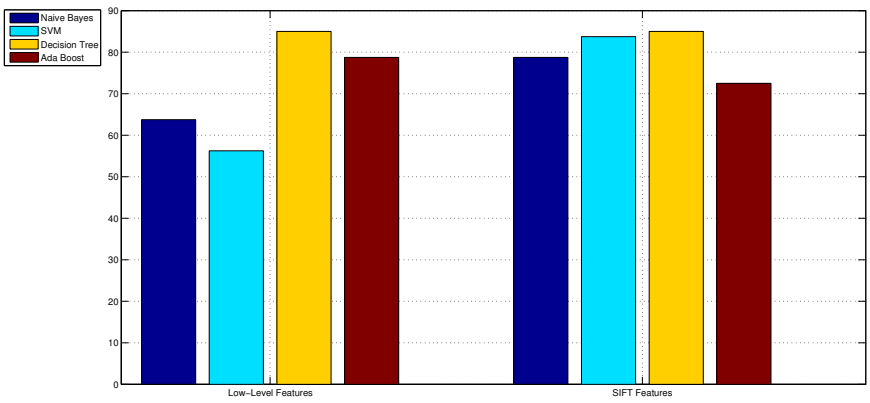

Figure 5.10: Best prediction results for the Blog classification using **720 images** for training and **80 images** for testing of each category with four different classifiers, using SIFT and Low-Level features.

5.3 Computational Study

The methods for select the most important features of a vector aims to eliminate features that aren't relevant for the classification of the web pages and without them the accuracies of the classifiers can improve. However, as it was shown in subsection 5.1, sometimes all the features can be important.

The increase of the number of features also increase the computational cost and the time that it takes to be made the training of the classifiers. As more features are used, more time is necessary and the results are not always the best.

In the SVM classifier using seven categories when the classification was performed using only the text contained in the web pages using half of the number of features the results were better. However, when the text features was merged with the visual features, the reduction of the number of features seems to decrease slightly the accuracy. As the table 5.1 shows, when it is performed the reduction for half of the number of features the time of training reduces over half (a relation that is approximately linear).

Table 5.1: Table of the training and testing time (in seconds) for different number of features with the SVM classifier.

Number of features	Training Time	Testing Time
14386	66,39	0,0584
13720	62,0183	0,055
7193	27,7723	0,0293489
3596	13,8318	0,0144875
2000	7,53474	0,00775625
1000	3,58964	0,00412337
666	2,40213	0,00271707
500	1,72942	0,0020434
333	1,06654	0,00143338
166	0,495024	0,000810144

In the second experiment, using four categories, the accuracies never decay bellow the 90% when the number of features are reduced. The time of training and testing have a similar behavior than with seven categories.

With the Decision Tree and Ada Boost classifiers the time of the training and the testing process are similar to that observed in the SVM classifier.

The Naive Bayes classifier needs very high resources to develop the train. As said before, all the tests using this classifier were done with a maximum of 2000 features. As it can be seen in table 5.2, with the available resources it was impossible to use all the features. The time of the test and training was too high, when compared with the other classifiers (a relation that is non-linear, with an exponential-like behavior).

Table 5.2: Table of the training and testing time (in seconds) for different number of features with the Naive Bayes classifier.

Number of features	Training Time	Testing Time
2000	20000.1	6.31906
1000	134,421	0.588251
666	35,041	0,2442
500	13.0473	0.124071
333	2.92052	0.0474861
166	0.329574	0.0130176

5.4 Overall Assessment

As observed in section 5.1, the visual features despite of obtaining bad results when used alone, contain useful information that improves the web page classification that was performed using only text. It was obtained good classifications. In some classifiers the accuracy was higher than 90%.

With seven categories, using the Ada Boost and Decision Tree classifiers the results were poor, never passing the 60% of accuracy.

However, it was obtained high accuracies when used the SVM classifier. With this classifier the results were good, passing the 90% of accuracy in almost all the combinations of type of features. As it was shown in section 5.3 when it is reduced to half the number of features, the time necessary to train the classifier decreased. The largest decrease of accuracy, when used half of the number of features, is 4%. When a better efficiency is needed and it is necessary a faster response of the classifiers the use of the half of the features in this classifier does not seems to be a deterrent.

With the Naive Bayes classifier it was obtained good accuracies, sometimes near to 90%. In both experiences, it have been achieved good results, always better then the accuracies of the Decision Tree and the Ada Boost classifiers. When used seven categories, the difference between this classifiers and the Naive Bayes was even higher than 30%. Bellow the 2000 features the accuracies tend to get worse. Maybe using more features it would be possible to get closer results

than those obtained with the SVM classifier. However, the training time of this classifier was too high.

As expected, with four classes the results were better then using seven classes. In the second experiment, all the classifiers obtained high accuracies, always above the 80% of accuracy. The computational times were lower then those obtained with seven categories. However, the training time is still high and the reduction of the number of features to half does not induce a decay of the accuracy too high. When it is needed a faster performance of the classifiers it can be reduced the number of features to half without loosing a considerable amount of accuracy.

Chapter 6

Applications

In this work it was performed several tests using the data set previously created. To train it was used 90% of the web pages and to perform the tests it was used the remaining 10%. Focusing in the web page classification by their topic, as stated before, it was used 450 web pages to perform the training and 50 web pages to perform the testing of the classifiers, for each category. As shown in chapter 5, it was obtained good results in the web page classification by topic using the combination of visual and text features. It was performed several tests varying, randomly, the web pages used to train and to test. The number of web pages in each category was fixed. The web pages were the same in all the tests.

Due to the explored above, we developed a web application, so that we can observe the behavior of our work when tested with other web pages. This application also allow us to create a larger and a dynamic data set, using the information provided by the users. This application can be tested by users that don't know how the program works and what were the web pages used to train the classifier.

As can be seen in chapter 5, the best accuracy obtained in the web page classification by topic was using the SVM classifier, when mixed the text features with the low-level features using 100% of the number of features. As stated in section 5.3, the time to train the classifier depends on the number of features used to describe each web page. However, the opencv has a function to save the training of the classifier. This function allow us to use the total number of features, since the training of the classifier will not increase the execution time of the application considerably.

To train the classifier, it was used the 500 web pages of each category of the data set. The

number of words increased, since it were used more web pages that contained other words. The application has no restrictions of the language of the web page. However, the web pages used to train are all in English.

This web application was created using small C/C++ scripts and implemented in the web using html and php. As the figure 6.1 show, in the homepage it is displayed the available categories for the classification of a web page. The user need to introduce the url of the web page that aim to classify. The classification process take some seconds, since it needs to extract the visual and the text features of the web page.

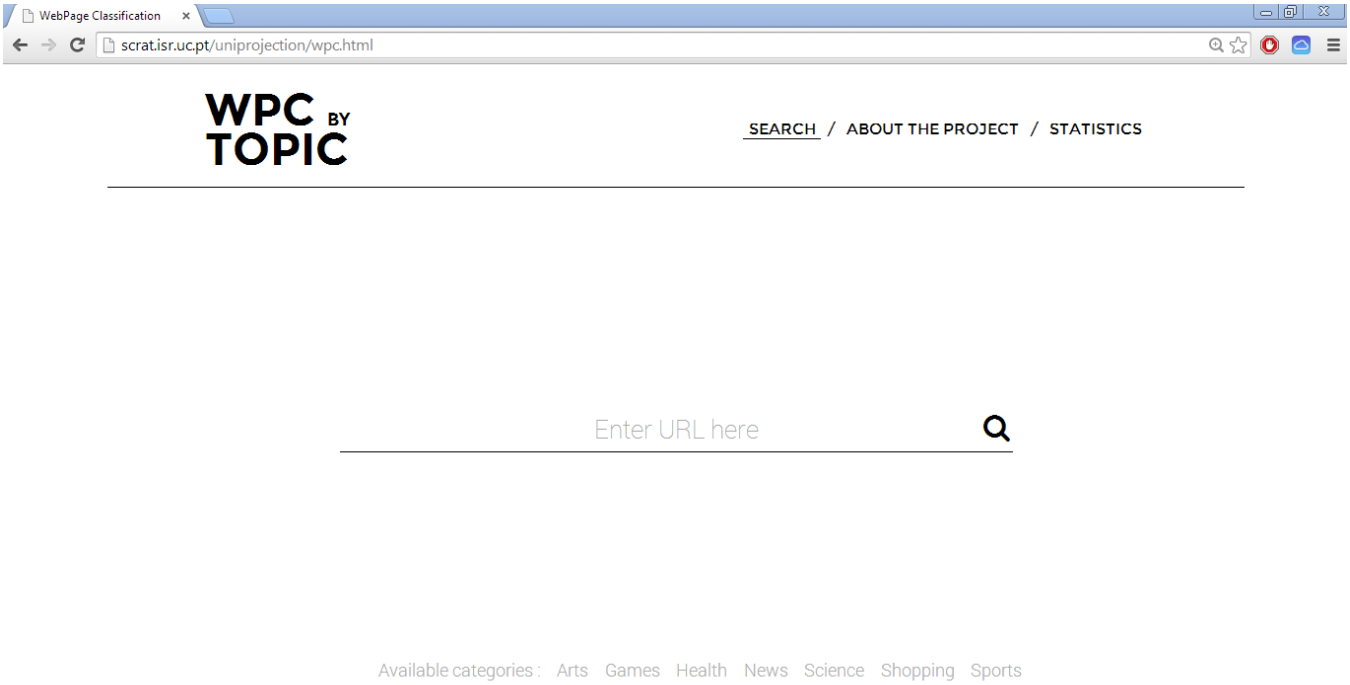

Figure 6.1: Print screen of the homepage of the web application.

After perform the classification, the result is displayed as the figure 6.2 shows. Since that one of the objectives of this application is the expansion of our data set, after the classification of a web page it is asked to the user if the classification was correct. If the answer is incorrect, the user can choose the correct category. When the application cannot render the web page and the number of words of the web page is less than 50, the classifier don't classify the web page. This application is available in "http://scrat.isr.uc.pt/uniprojection/wpc.html" and, so far, it were classified 36 web pages with an accuracy of 91,67%. It is also possible to visualize the statistics of our application clicking on the tab "Statistics".

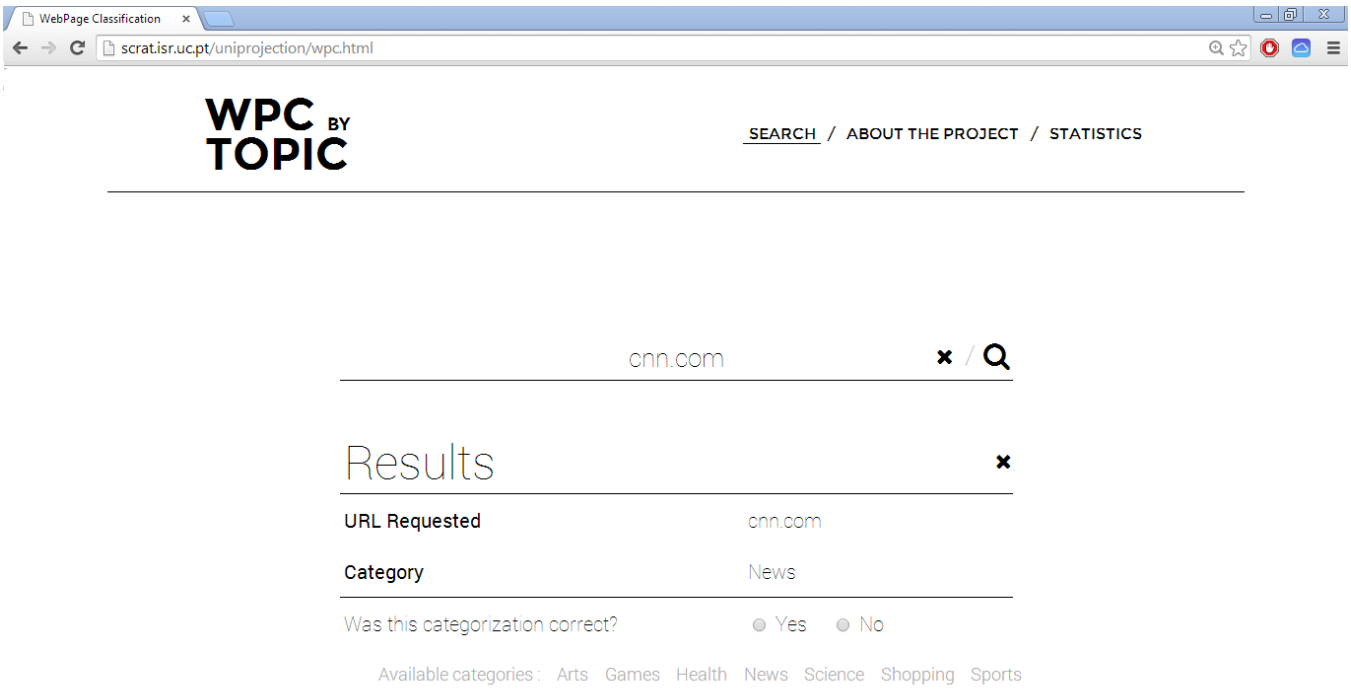

Figure 6.2: Print screen of the web page that shows the result of the web page classification.

Chapter 7

Conclusion

In this work we described an approach for the automatic web page classification by topic using the combination of text and visual features. We also described an approach using visual features to do a binary classification of web pages to distinguish if they are a blog or non-blog. It was obtained good results. They show that, in fact, generally the visual features improve the classification of the web pages and should not be ignored.

Our approach to categorize the web page by topic is based on text features extracted from the HTML file of the web page merged with the low-level and SIFT features extracted from the visual content of it. It was made all the possible combinations between the visual and text features. It was also used the Chi-Square method to select the best features of the vectors to try to improve the results.

To distinguish between a blog and a non-blog web page it was only used the visual features. In this binary classification it was obtained high accuracies, with all the classifiers, mainly using SIFT features.

We also present a computational study to understand the costs of the increase of the number of features.

To present our work we also developed a web application that can be experienced by any person.

Our results come to reassert that in multi-label classification problems more classes tend to decrease the accuracies. With more topics the difference of visual characteristics between them become smaller. The web pages begin to be confused to the classifiers, since that the increase of

topics also increases the common characteristics between them. This happens when the size of the data set don't change. However, despite the low results when used alone, the visual features improved the classification of the web pages when mixed with the text features.

Although the accuracy of the classifiers improved when the visual features are added, the improvement is not always big. This may be due to the huge difference between the number of visual features and the number of text features used. The visual feature universe is too small when compared with the text feature. More features can be added, in future work, to the low-level features descriptor and more SIFT descriptors can be extracted from the web pages to mitigate this difference. It was, however, expected that the visual features of the web pages didn't have much information about the topic, since the categories used in this work are too general.

This can be the fact that explain the poor results of the classifications by topic when used only the visual features. Increase the number of web pages to train by category may improve the accuracies. Other option is to create a data set with categories not so generalist.

In our work the HTML structure of the web page was ignored. As described in chapter 2 some works show that the structure may have some information to help to classify the web pages. As the objective of this work was to show that the visual features can help to the web page classification, that approach was more complex but can be used in future work to improve the results.

In this work it was used the Chi-Square method to select the top features of the vectors. As can be seen in chapter 5, the accuracies sometimes improved when used the chi-square method. In future work another feature selecting methods can be tried.

Sometimes, a web page can be classified with more than one topic (i.e. there are too many Sports news web pages). This fact can lead to misclassification of some web pages. A possible approach is to calculate the correlation between all the topics and the web page using a regression technique. It would also be possible to achieve a matching degree between each web page and every topic. In this way it is possible to understand if one web page may belong to more than one different category.

One important issue to classify a web page using the text is the language. In our approach it were only used web pages written in English. Since it is used an approach that don't consider the order and the structure of the text (bag-of-words), like in our work, a solution is to preprocess web pages, doing their translation to English. But this raises other problems because there are

too many different languages.

As we said above, in this approach we didn't give any importance to the structure of the text. Sometimes this can be considered a problem. Some words are directly connected to others and using the bag-of-words each word is considered independent. For example "water closet" is analyzed as "water" and "closet" without any connection. These words, when analyzed independently, don't have the same meaning that when they are together.

Using the words without any analysis could lead to a worse accuracy. Words like "movies" and "movie" should also be used as one unique word. In other works that use the text to do the web page classification, they used some stem algorithms to improve the accuracies of their classifiers. However after a search we tried to use a recent method based on Porter's algorithm [13] and some words are incorrectly stemmed. The result of the stem using the words "movies" and "movie" is "mov". We found more examples like this and decided not to use the stem algorithms. This don't have a big importance to the objective of this work.

Additionally, using the term frequency-inverse document frequency method, words that are contained in all the documents (stop-words) have less importance that rare words. However, this words can have influence in the accuracies of the classifiers. In previous works this stop words were removed "manually". But nowadays there are lists of words that are considered "stop words". An improvement that can be done in our work to improve the accuracies of the classifiers is use these lists and remove them from the dictionary.

As it was said in chapter 4 to reduce the number of words of the "bag" , it was filtered the words that were present in all the documents and that were present in less then 10 documents. It was not used any scientific criterion to do this selection. However we believe that with all that words that was filtered the results would be worse.

In our work we merged the features (text, low-level and SIFT), with all the combinations between them, and it was performed the training and the test of the classifiers with one vector of features for each web page. Another approach could be training and testing the classifiers with each type of features and then use a vote scheme as in [18].

The web pages contain some text that are linked to other web pages. In our approach it was used only the text of the documents, ignoring the linked web pages. In future work it can be useful to use the information contained in the linked web pages to help to find the web page topic.

As stated before, blogs are being used more and more. It was obtained high accuracies in the binary classification to distinguish if a web page was a blog or non-blog. In this classification only visual features were used. In future work it can be made the fusion between the visual features and the text features contained in the blog. It can also be made a classification of the blogs by topic, using the same strategy as implemented for the web page classification by topic.

The web application developed is simple and more work can be developed. One of the problems of our application is that only web pages in written in English were used to train the classifier. The text features, when is introduced a web page written in other language, are useless. As said before, can be developed a strategy to translate the web pages to English before the extraction of the text features. Our application have only the seven categories used in this work. In future work it can be increased the number of categories. After the application ends, it is asked to the user if the classification was correct. With this information, it can be increased our data set and in future work we can develop a strategy to train the classifier automatically with the new web pages introduced.

The results obtained in this work were very positive and shows that in fact the visual features complement the web page classification by topic that uses only text features. It shows that when used alone, these features are strong mainly for subjective variables, such as Gonçalves and Videira [23] already concluded. More types of subjective variables can also be developed, such as to distinguish if a web page transmits happiness or sadness for instance. The application developed can be improved and may be added new types of classification.

References

[1] Arul Prakash Asirvatham and Kranthi Kumar Ravi. Web page classification based on document structure. In *IEEE National Convention*, 2001.

[2] Christopher H Brooks and Nancy Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th international conference on World Wide Web*, pages 625–632. ACM, 2006.

[3] Rung-Ching Chen and Chung-Hsun Hsieh. Web page classification based on a support vector machine using a weighted vote schema. *Expert Systems with Applications*, 31(2):427–435, 2006.

[4] Viktor de Boer, Maarten van Someren, and Tiberiu Lupascu. Classifying web pages with visual features. In *WEBIST (1)*, pages 245–252, 2010.

[5] Alexiei Dingli and Sarah Cassar. An intelligent framework for website usability. *Advances in Human-Computer Interaction*, 2014, 2014.

[6] Nuno Gonçalves and Antonio Videira. Automatic web page classification using visual content. In *International Conference on Web Information Systems and Technologies*. WEBIST, 2013.

[7] Min-Yen Kan and Hoang Oanh Nguyen Thi. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326. ACM, 2005.

[8] Milos Kovacevic, Michelangelo Diligenti, Marco Gori, and Veljko Milutinovic. Visual adjacency multigraphs-a novel approach for a web page classification. In *Proceedings of SAWM04 workshop, ECML2004*, 2004.

[9] Oh-Woog Kwon and Jong-Hyeok Lee. Web page classification based on k-nearest neighbor approach. In *Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, pages 9–15. ACM, 2000.

[10] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, pages 388–388. IEEE Computer Society, 1995.

[11] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[12] Maryam Mirdehghani and S Amirhassan Monadjemi. Web pages aesthetic evaluation using low-level visual features. *World Academy of Science, Engineering and Technology-49*, 2009.

[13] Martin F Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.

[14] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[15] Daniele Riboni. *Feature selection for web page classification.* na, 2002.

[16] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[17] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[18] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, and Wei-Ying Ma. Web-page classification through summarization. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 242–249. ACM, 2004.

[19] Aixin Sun, Maggy Anastasia Suryanto, and Ying Liu. Blog classification using tags: An empirical study. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 307–316. Springer, 2007.

[20] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, 1978.

[21] Shawn C Tice. *Classification of Web Pages in Yioop with Active Learning*. PhD thesis, San José State University, 2013.

[22] Ms Sonal Vaghela, Mr MB Chaudhary, and Mr Devendra Chauhan. Web page classification using term frequency. *International Journal For Technological Research In Engineering*, 1:949–954, 2014.

[23] Antonio Videira. Web page classification using visual features. Master's thesis, University of Coimbra, 2013.

[24] Alexa website. Alexa - actionable analytics for the web. http://www.alexa.com/, 2012.

[25] Elinks website. Elinks - full-featured text www browser. http://elinks.or.cz/, 2002-2008.

[26] Wahyu Wibowo and Hugh E Williams. Simple and accurate feature selection for hierarchical categorisation. In *Proceedings of the 2002 ACM symposium on Document engineering*, pages 111–118. ACM, 2002.

[27] Michifumi Yoshioka. Web page classification method using neural networks. *IEEJ Trans. EIS*, 23(5), 2003.