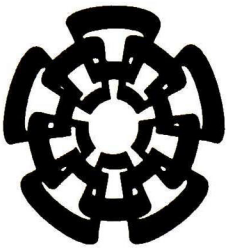
CT-865-551
DOV. 7015



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Reconocimiento de sentimientos y emociones en el análisis de la web

Tesis que presenta:

Fernando Santos Sánchez

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Director de Tesis

Dr. Andrés Méndez Vázquez

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Noviembre de 2014.

Reconocimiento de sentimientos y emociones en el análisis de la web

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Fernando Santos Sánchez
Ingeniero en Comunicaciones y Electrónica
Universidad de Guadalajara 1998-2002

Becario de CONACYT, expediente no. 476638/283060

Director de Tesis
Dr. Andrés Méndez Vázquez

CINVESTAV del IPN Unidad Guadalajara, Noviembre de 2014.

Ohana means family.
Family means nobody gets left behind, or forgotten.

— Lilo & Stitch

This work is dedicated to my family, for you are my all.

RESUMEN

En la actualidad se ha hecho indispensable el uso de sistemas capaces de encontrar la información relevante a una consulta dada en la inmensidad de la Web. Dichos sistemas se denominan motores de búsqueda de los cuales Google es el más famoso. Si bien dichos motores dan buenos resultados en cuanto a tiempo de respuesta, relevancia y popularidad, la implementación de sistemas con mayores capacidades se ha vuelto una necesidad creciente. Ejemplos de dichas capacidades es la categorización de los resultados por el tipo de página, el análisis del contexto de los resultados, la posibilidad de comparar consultas similares, la identificación de grupos a favor y en contra, y la caracterización de sitios de riesgo.

En esta tesis se busca dar los fundamentos para la creación de dichos sistemas mediante el uso de técnicas de reconocimiento de emociones y sentimientos. Partiendo de la idea de que la mayor parte de las opiniones se relacionan con emociones y sentimientos, en este trabajo se busca segregar los resultados aportados por un motor de búsqueda en base a su contenido emocional/sentimental. Dicho agrupamiento permitirá formar una caracterización de los diversos tipos de páginas, interpretar la opinión de la comunidad acerca de la consulta, y encontrar sitios a favor y en contra.

ABSTRACT

The data in the Web has increased to the point that to access it, the use of specialized systems are needed. These systems are commonly named search engines, where Google is the most famous and employed. The focus of the search engines is to provide the user with a list of the most relevant results to a query in the minimum time required with no human intervention during the process.

Even with the most efficient of the current systems, It is quite apparent the necessity to design and implement new capabilities. For example, topic categorization, page context analysis, opinion identification and characterization of risk groups are tasks that could greatly impact in the user experience.

Most of the previous examples can be related to finding the opinions expressed for each result. Commonly, opinions are associated with emotional or sentimental expressions. This allows the identification and characterization of diverse groups by categorizing them into communities with similar emotional patterns. Furthermore, the emotional content present in the communities can be analyzed to conform the community current opinion. Thus, with an interface implementation and further process refinement, the proposed system could allow the comparison between similar queries by identifying the common emotional response and generating executive reports for the user.

PUBLICATIONS

Some ideas and figures have appeared previously in the following publication:

Santos, Fernando and Mendez-Vazquez, Andres "Sentiment Identification for e-Services," *in International Conference on Advanced Applied Informatics* (IIAI AAI 2014), 5th International Conference on E-Service and Knowledge Management (ESKM 2014).

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth [22]

ACKNOWLEDGMENTS

I give my deepest thanks in first instance to my professor Dr. Andres Mendez-Vazquez, who gave me the guidance to successfully realize this work while at the same time procured my research independence without any impositions. To my family who supported me, in this hour I tell you this, you are an essential part in I my every day, and you are always in my heart. To the close collaborators which came together to gave counsel and assistance in this project, I give you the recognition you rightly deserve:

To Saif Mohammad for his commitment to improve the computer science area by granting access to his research results[1] Without a emotion lexicon this work would have been imposible to realize.

The brothers Erick Ivan Sanchez & Carlos Omar Sanchez for your diligent work in the design of the first crawler module, I know your work was especially difficult because you did not have previous experience in the area and the resources where not always the best. My best wishes to you, and please never stop trying.

To Tania Rivas for her contribution in the testing & validation phase, Your works permitted to simplify the code by finding and correcting not only errors, but even whole process and structures. You where the backbone in the last five versions of the program and for that you will always have my appreciation.

To Tania Rivas for her design and implementation of the Web interface, an admirable work which complements the system to an almost commercial level.

Also I would like to mention my personal friend Arturo Ismael Quinto who provided the motivation and advice in those times of necessity when the program refused to run with out explanation. As a final instance I would like to acknowledge the institutions Cinvestav and CONACYT because they provided the necessary resources to create this work.

1 http://www.purl.org/net/NRCemotionlexicon

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

"Truth is ever to be found in simplicity, and not in the multiplicity and confusion of things."— Frank E. Manuel [28]

PROBLEM DEFINITION

The purpose of this works is the design and implementation of a system capable of analyzing the Web data of an heterogeneous page set. Giving that in most cases, pages from different sites do not have a common scheme or organization, this system required another type of metric. The selection of the sentiment and emotion detection metrics obeys to the personal conviction that every human task involves emotions. Here, the methods and techniques to process and analyze text and documents put together the emotion metrics and clustering algorithms to conform the base of this new system in Web Analysis.

In particular this system will operate in the following way:

Obtaining a initial page set related to a query made in a Web search engine.

- Representing the data stored in each page by using a word vector.

Clustering the pages into page communities with similar emotional output.

- Graphically representing the data by using graph techniques.

Interpreting the data in a simple report for the user.

Each one of these tasks is done by one or more system modules, and each of them will be further explained in the next chapters.

PROBLEM RATIONALE

Giving the enormous quantity of data in the Web, specialized systems to search, organize and analyze the data, have become a necessity. The most commonly employed system in this area is the Web search engine whose processes are explained in(Sec: 3.2). While the use of this kind of systems have become generalized, many problems still remain and new ones are appearing. Modern day technology needs a way to transform and represent the Web data into structures that facilitate their processing. The Web data have two important aspects that must be observed differently, the data present in the Web page and the relationships between the pages. It is common enough to use graphs to represent the latter aspect, where many approaches have appeared to quantify and analyze them. While the research of the the most relevant works in this area is briefly approached, this thesis focus on the Web page data.

It is our hope to contribute in the development of a new systems designed to consider one of the principal human qualities, the emotion expression. This can be accomplished by designing a system that can analyze an initial set with hundreds of thousand pages by representing each page into a vector form, and clustering the pages based in their emotional similitude.

Part II

STATE OF THE ART

"It ain't the guns nor armament, Nor funds that they can pay, But the close co-operation, That makes them win the day.It ain't the individual, Nor the army as a whole, But the everlasting team-work Of every bloomin' soul."— Mason J. Knox [21]

3

DEFINITIONS & METRICS

In this chapter, the definition of a selection of metrics and terms is given, and while some of the approached topics may appear to be basic to the knowledgeable, this term selection seek to provide the reader with all the tools to comprehend the rest of this work without the presumption of previous knowledge in the area. Therefore, this section is intended to be used as a reference and to provide a comprehensive list of related articles for the reader perusal.

The definitions and metrics in this section can be grouped into two areas, Graph Theory and Web Analysis. For each area, the most relevant information is provided covering not only the topics approached by this thesis, but also the related works where some of the ideas arose.

3.1 GRAPH THEORY

Most of the definitions in this section come from the text book "Graph Theory with Applications" [3]. Thus, any course in graph theory will give the reader enough information to understand the topics approached by this work. The drafting of this section obeys the desire to give the reader all the required knowledge to fully understand this work without making any assumption about previous knowledge about the topic. As a first step let give the reader a formal definition of a graph.

Definition 1. *A formal description of a graph can be given as a triplet*

$$G = \{V(G), E(G), F(G)\},$$

where $V(G)$ is a set of vertices, $E(G)$ is a set of edges, and $F(G)$ is an incidence function that associates each edge from $E(G)$ with a pair of vertices from $V(G)$. Where $V \bigcap E = \emptyset$.

The following the previous definition take us to the concept of vertex. A vertex can be better understood as the graphical representation of one of the elements being studied, normally also called a node and it is most often detailed as a small bullet in the graph. Also, the description of the relationships between vertices is primordial. They are normally represented as edges, and this relationships can represent any number of things from the distance between two cities to a link between two Web pages, where the cities and the Web pages would be represented by the graph nodes. This relationships are normally

seen as association functions between nodes. The most common association functions are the incidence and the adjacency relationships, which are explained in the following definition:

Definition 2. *Given a graph* $G = \{V(G), E(G), F(G)\}$, *we have the following concepts:*

1. *The vertices* $v_i, v_j \epsilon V(G)$ *at the end of a edge* $e_k \epsilon E(G)$ *are said to be incident to the edge and vice versa.*
    a) *Two vertices* $v_i, v_j \epsilon V(G)$ *are adjacent iff there exists an edge* $e_k \epsilon E(G)$ *that is incident to both.*
    b) *The degree of a vertex* v_i *is defined as the total number of edges incident to* v_i. *In directed graphs in-degree is the total number of input edges to* v_i, *and out-degree the total of the output edges.*

Given these essential concepts of the graph structure, the following concepts will provided a way to categorize graphs. First, between this concepts is the definition of a loop:

Definition 3. *A loop is generated when both vertices associated by an edge are the same. A walk between a pair of vertices is an alternating succession of vertices and edges, where every vertex in the sequence is adjacent to the preceding and following edges.*

Many graphs of interest are simple graphs, a simple graph can be defined with the previous concept as:

Definition 4. *A simple graph is one that includes no loops or different edges that associate the same pair of vertices. A graph is connected if exists a walk between any pair of vertices in the graph. In a directed graph each edge is associated a direction, a similar case occurs with weighted graphs.*

An specific example of simple graphs is the tree family which is a powerful tool in the area of computer science for their capacity in creating economic structures to store and access data, in addition to representing the computer science concept of recursion. Another important tool is the matrix representation of a graph. While a graph can help to clearly visualize the internal relationships of a problem and to condense great amounts of information, most computer programs access this information by using a matrix representation and no a graphical one. The most common matrix representations are the incidence and the adjacency matrix formally defined as:

Definition 5. *For an incidence matrix* $M(G)$, *the columns represent edges and the rows vertices, each value* m_{ij} *in the matrix is given by the number of incident edges* e_i *in a vertex* v_j (0,1, or 2). *In the adjacency matrix* $A(G)$ *both columns and rows represent vertices where each value in the matrix* a_{ij} *is given by the number of edges that join the vertices pair.*

Also, because many works employ the idea for implementing clustering algorithms, the definition of clique must be given:

Definition 6. In an unweighted graph G, a clique is a sub-graph H where each pair of vertices, $v_1, v_2 \epsilon H$, is connected by an edge.

To end this section, the reader is presented with the most common representation of the Web in the works studied here.

3.1.1 *The Web as a Graph*

One of the most relevant applications of graph theory for this work is the web graphical representation [6]. The web can be represented as a graph where each vertex represents a page, and the edges represent links between pages. Based in the information provided by such graph, and its adjacency matrix, it is possible to develop models for Web Search Engines [43, 36, 20, 19, 14], Phishing Automatic Detection [27, 26, 46], Communities Detection [39] and Plagiarism Detection [37, 7].

3.2 WEB ANALYSIS

The second area to be studied is Web Analysis. In this work, Web Analysis refers to the set of techniques, methods and tools to extract and comprehend the Web information. The most common perception between the Web users is that the Web pages text conform the principal source of data in the Web, while is true that the page text is an important part of the data, the previous idea ignore many other sources like the pages structure, the linkage data or the different kinds of content stored by the page. Many of this topics are research areas for the Web Analysis, and in this section the reader is given concepts of the most common approaches to Web Analysis: The Web Text Analysis and the Analysis of the Web structure.

In this thesis, the Web Text Analysis conforms the central part being necessary to identify the emotions and sentiments displayed in each page. The second area, when combined with the data obtained in the first steps, together with the Web text analysis, can provide new opportunities for more intensive techniques. Example of this would be to refine the clustering process from a total page set to the clustering of site pages giving the user a detailed information about the relations between sites.

3.2.1 *Text Analysis: Concepts, Definitions & Metrics*

Most of the approaches for Web Text Analysis are direct implementations of Text Analysis methods and techniques. Therefore, in this

section, the principal concepts and metrics employed in Text Analysis are given. Many of these were not employed in a direct manner in this thesis, but served as inspiration for the proposed process of sentiment and emotion identification implemented during this work. Thus, first, a series of concepts are defined making possible to explore the most common metrics currently in use.

While some of the early approaches sought to analyze all the text, word for word or even letter by letter, most of the current ones characterize a document in a way that makes it easier to handle through special representations. In this section, the concepts of two approaches are given: the conformation of shingles, and the representation by a page vector. The metrics employed by each approach are described in each of the following parts.

3.2.1.1 *Shingles*

For the shingles [7] approach two simple concepts are needed.

Definition 7. *A token can be seen as a contiguous sequence of words, letters or lines. A contiguous sub-sequence of tokens contained in a document, D, is named a shingle and It is represented by the function $S(D, w)$, where w indicate the shingle size.*

With the previous definition, the idea of representing a document by shingles can be visualized as follows: Be document C the original by defining the tokens as words, the document is broken into a set of shingles of n-length. Therefore, each shingle can be seen as a set of words, and depending from the method employed repeated words can be eliminated. This process ignores all the non-words elements in the document permitting to transform it into a canonical form, which means that any two documents that differ only in formatting will be reduced to the same set of shingles.

Once the shingle set of a document is obtained it is possible to define the containment and the resemblance between a pair of documents.

Definition 8. *The resemblance between a pair of documents can be measured by the rate of common shingles over the total number of shingles in both documents. In a similar way the containment of document A in another will be the number common shingles over the total number of shingles in document A.*

Once the concepts for these topics are given, it is time to present the metrics employed in this area. As a first metric, the resemblance between two documents can be expressed:

$$r(A, B) = \frac{|S(A, w) \cap S(B, w)|}{|S(A, w) \cup S(B, w)|}, \tag{1}$$

Also, the containment of a document A into B is defined as:

$$c(A,B) = \frac{|S(A,w) \cap S(B,w)|}{S(A,w)}. \quad (2)$$

Even when the two previous formulas are useful for Web Analysis process for finding plagiarism in the Web [37, 27, 7], it is important to note that the formula for the resemblance of two documents is not a proper metric because it does not obeys the triangle inequality. To solve this problem, a new modification must be made to the formula, thus defining a metric:

$$d(A,B) = 1 - r(A,B). \quad (3)$$

3.2.1.2 *Page Vectors*

For the second idea, We believe that any document can be represented as a vector where each component stores the times a certain word appears in the page. While by itself this can be unmanageable, when properly delimited it can become an useful tool. Instead of counting each word as a separate entity each vector component can store the number of words pertaining to a certain aspect present in the page. Even more, by dividing the value of each component over the total number of page words, the percent of the text related to an specific aspect can be calculated. This thesis uses the last type of vector denominated as a percentile emotion vector to characterize the total page set. The principal idea is to find the percent of related text to any emotion of sentiment category studied.

To better understand the previous words let introduce a fundamental part of this thesis, the emotion Lexicon[32]. An emotion lexicon can be seen as an especial kind of dictionary. For this consider that words can be associated with emotions or sentiments. For example, both yummy and delightful express joy while cry may indicate sadness. So, in the work "NRC Emotion Lexicon" [29] a dictionary that stores the the emotional and sentimental relationships of each word instead of their meaning was created. This lexicon is freely provided for research purposes at the official site [29].

Each element in the lexicon can be seen as a vector, which stores the word and its relation to each sentimental and emotional category. The relationship of a word with the $\imath th$, $i = 1, 2, ..., n$, emotional or sentimental class can be represented formally by the indicator function$\delta_w(\imath)$, where $\delta_w(\imath) = 1$ indicates that the word evokes the emotion or sentiment, and $\delta_w(\imath) = 0$ represents the opposite.

In (Sec. 4.2) will give a more in deep detail of both the process to conform an emotion lexicon [32], and diverse works [30, 33, 31] that employ this lexicon to perform a sentiment analysis.

3.2.1.3 *The Levenshtein Distance*

The Levenshtein distance [24] can be visualized as a measuring of the difference between two sequences of characters, i.e. the minimum number of single characters edits needed to transform one character sequence into the other.

Commonly denominated the edit distance, it was discovered by Vladimir Levenshtein in 1965 [[24]], and it is mostly employed when working with string alignments between character sequence pairs. The recursive function is defined as:

$$lev_{a,b}(i,\ j) = \begin{cases} max(i,\ j) & if\ min(i,\ j) = 0, \\ min \begin{cases} lev_{a,b}(i-1,\ j) + 1 \\ lev_{a,b}(i,\ j-1) + 1 \\ lev_{a,b}(i-1,\ j-1) + [a_i \neq b_j] \end{cases} & Otherwise. \end{cases} \tag{4}$$

Given the single character edits (insertion, deletion and substitution), establishing the upper and lower bounds of the function is possible: The Levenshtein distance is at most the length of the longer string, and at least the difference of the sizes of the two strings. For example, the Levenshtein distance between kitten and sitting equals 3 necessary edits, while the distance between pepsi and coke is 5, and the distance between cake and coke is 1.

3.2.1.4 *The Hamming Distance*

The Hamming distance [13] is a metric used with strings pairs of the same length, and it is defined as the number of positions in which the corresponding symbols are different. Compared with the previous metric, it measures the number of substitutions required to change one string into another of the same length. It can also can visualized as the minimum number of errors that could have transformed one string into another.

For a fixed length n, the Hamming distance is a metric on the vector space of the words of length n, and can be defined in a similar way to the Levenshtein distance. As an example the Hamming distance between "toned" and "roses" is 3, given that the letters in the 1°,3°, and 5° positions are different, so to change one string into the other we will require to replace three symbols. Another way to see the previous example is that after transmitting the first string the minimum number of possible errors at the reception point is 3.

3.2.1.5 *The Jaro-Winkler Distance.*

The Jaro-Winkler distance [16] was designed to measure the similarity of two strings and it is classified as an edit distance type. It is given by

a continuous function in the range of [0,1], where the function score is normalized such a value of "1" implies an exact match, and a "0" value no similarity at all. It is used the record linkage area to detect duplicates because it is best suited for the comparison of short strings. This distance counts the number of matching characters but also includes the idea of transposition. The characters that occur in both strings but are not in the same position become transposed characters, being the number of transpositions equal to difference between their positions. Then, the Jaro distance can be mathematical defined as:

$$d_j = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right) & otherwise \end{cases} \tag{5}$$

where m is the number of matching characters and t is half the number of transpositions. Two characters are considered a match if they are the same and not farther than a number of transpositions. The number of transpositions for two strings s_1 and s_2 is given by the following formula:

$$\left\lfloor \frac{max(|s_1|, |s_2|)}{2} \right\rfloor - 1. \tag{6}$$

The Jaro-Winkler distance employs a prefix scale to favor strings that match the beginning of a set prefix with length ℓ. Thus, it is possible to define the Jaro-Winkler function as:

$$d_w = d_j + (\ell p(1 - d_j)), \tag{7}$$

where d_j is the Jaro distance for strings s_1 and s_2, ℓ is the length of the common prefix up to 4 characters, and p is the constant scaling factor for how much the score is elevated by containing common prefixes. The value of p is commonly 0.1 and it should not exceed 0.25. Also, it is important to clarify that this distance does not complies with the triangle inequality.

3.2.2 *Structural Web Analysis: Concepts & Definitions*

The Web Analysis has incorporated techniques and methodologies [27, 26] to comprehend the structure of both the pages and the Web itself. While most approaches focus on the Web structure, this section also comments about the use of the visual structure of a page which can be useful attaining and representing new data in the nodes of a typical Web graph. For the Web structure analysis most works depend on building a crawler to gather the linking data that represents the relationships between the pages of a network.

3.2.2.1 *Crawling in the Web*

Given the most common idea to visualize the Web as graph where the pages are represented by nodes and the edges represent links, the Web crawler [12] has become an almost essential tool to perform any Web Analysis. As such, the crawler programs are highly diversified as they are produced with a wide range of necessities in mind. Here, it is given a set of specific task that a general crawler should perform independently of its configuration:

To enter a initial page.

To obtain all links in the page.

To store the linking data and to repeat the process with all the linked pages.

Most of the time the linking data is stored as adjacency or incidence matrix, and each page is given an identifier for easier operation. When one page has more than one link to another one, some procedures add a "weight" to the edge representing the link, this weight is equal to the number of links from page A to page B. Even if the process itself appears simple, many considerations should be made to avoid an ample range of problems like the cyclic nature of the Web, the algorithms processing time and even the technical legalities of accessing and indexing the Web pages [1]

3.2.2.2 *Page Structural Analysis*

While most works in the area focus in gathering the linking data of the Web or sub-net to conform a structure analysis it should be noted that the page structure itself could provide useful insights that should be taken into account. Many elements apart from the text can be taken from a page, and the analysis of this elements can provide the data to solve diverse problems. Examples of these ideas are the works focused on finding phishing sites [27, 26] where a characterization of the each page is compared with the parameters of a target page. The most common characteristics employed for the structural analysis of a Web page are:

- Visual theme

  Page text

  Domains name

  Google assigned Page Rank

  Age of domain

1 http://www.bna.com/legal-issues-raised-by-the-use-of-web-crawling-and-scraping-tools-for-analytics-purposes/

Suspicious characters in the URL

- The appearance of known images.

The similarity between each characteristic is often determined by a metric dependent of the characteristic type. For example, the domain similarity will employ a text metric as the Levenshtein distance to compare an URL with the previously added to identify repeated links or links that try to emulate other pages.

3.3 CLUSTERING

3.3.1 *The Concept Definition*

Intuitively, clustering refers to the grouping of individuals or objects in a way that those in the same group/cluster are more similar with respect to a certain metric than to those of others groups/clusters.

Definition 9. Given a set of elements $S = e_1, e_2, ..., e_m$, a subset C is called a cluster if $\forall x, y \in C, \forall z \notin C \Rightarrow d(x,z) < d(x,y)$, where $d : S \times S \rightarrow \mathbb{R}$ is a distance function between the elements in S.

Once the concept of cluster has been defined, it is easy to see that clustering is not a method by itself, but a task to be solved. In its inception clustering was developed by Driver and Koeber in 1932 [23] in anthropology to evaluate the quantitative expression of cultural relationships.

Many approaches and metrics have been researched, with each area and application seeking to optimize the process for the data and objectives specific to it. Thus, given the general clustering definition, many interpretations of cluster have emerged and with each one a new model is generated to fit the particular interpretation. Most of them take the ideas from other areas to build a more specific definition of a cluster. In this work, we explain the most common ideas about clustering, and we focus on the Quasi Clique Merger Method (QCM) for clustering [35].

Most clustering models are based in the following ideas [11]:

The distance connectivity.

Graphs interpretation.

Statistical distributions.

- Density regions.

Members and attributes.

A mean vector.

Not to say that these are the only models, but they are the most common. In addition, a cluster can be categorized by the specific relationship between the cluster elements being the most common [44]:

- Each object belongs to exactly to one cluster.

  An object belongs to one cluster or does not belong to any cluster.

  An object belongs to a child cluster and to parent cluster.

  An object can belongs to any possible number of clusters.

After this, the Quasi-Clique Merger Method (QCM) in the following chapter is introduced [39, 35]. This method is employed by the present work to make the clustering of the pages based in their sentimental word density.

4

PREVIOUS RELATED WORKS

This chapter not only gives the reader a summary of the related works that where used directly in the proposal implementation, but it also includes those that gave valuable ideas and inspiration for the proposed work. These can be roughly divided into tree main areas: Web Analysis, Sentiment/Emotion Identification, and Clustering Algorithms. Each of them will be approached in the following sections.

4.1 WEB ANALYSIS

As in the previous chapter there are two strong variants in Web Analysis, to analyze the page text and to analyze the Web structure. This work seeks to combine techniques from both areas to expand the knowledge gathered by the process. For this, the chapter will cover concepts from the areas of text analysis, document characterization, plagiarism detection and structural analysis, which are employed in the Web search engine process, the influence detection in social networks, and the identification of phishing sites. In addition, some works that could be cataloged in the document characterization will be included in the section of sentiment and emotion identification.

4.1.1 *Text Analysis*

In order to implement the text analysis techniques for the Web analysis, it is necessary to consider the type of text to be analyzed. From analyzing short text like e-mails and tweets to the analysis of novels and literary works, it is important to characterize the document. Here, it is a summary about the characterization of a document, and an example of how to use the characterization to find phishing sites in the Web.

4.1.1.1 *Characterizing The Document*

In the previous chapter the concepts to conform the document characterization were given. Here, the process of shingle creation is given a closer look. For this, lets consider a brief example:

Given a document D conformed by the next poem its taken as an input:

```
Roses are red
Violets are blue
I've never found someone
```

```
as patient As you
```

It can be conformed a shingle set. First, each word its taken as a token and then each shingle is defined as set of 3 tokens, $w = 3$. So, the document D can be expressed as the next shingle set:

$$S(D,w) = \{\{Roses, are, red\}, \{Violets, are, blue\}, \{I, have, never\}, \{found, someone, as\}, \{patient, as, you\}\}, \quad (8)$$

As can be seen a shingle is an aggregation of contiguous tokens, this gives the programer a lot of room for personalization. Should repeated words be eliminated? Should contractions be expanded? all depends on the programer needs the only restriction is that the process be consistent.

4.1.1.2 *Document Characterization Application: Plagiarism Detection*

Once the concepts for this topic are given in the section it is time to present the metrics employed in this area, as a first metric the resemblance between two documents can be expressed:

$$r(A,B) = \frac{|S(A,w) \cap S(B,w)|}{|S(A,w) \cup S(B,w)|}, \quad (9)$$

Also, the containment of a document A into another is defined as:

$$c(A,B) = \frac{|S(A,w) \cap S(B,w)|}{S(A,w)}. \quad (10)$$

Even if the two previous formulas are useful for Web Analysis process as finding plagiarism in the Web [37, 27, 7], it is important to note that the formula for the resemblance of two documents is not a proper metric because it does not obeys the triangle inequality. To solve the previous problem a new modification must be made to the formula, thus defining a new metric:

$$d(A,B) = 1 - r(A,B). \quad (11)$$

4.1.2 *Structural Analysis*

4.1.2.1 *Web Structural Analysis App: Search Engine*

It is not a easy problem to make quality searches in billions of pages that compose the actual Web. Even more, before Page Rank, most of the selections were made by hand [43]. Given the clear disadvantages of low speed and a high error rate in the results, Google introduced the Page Rank algorithm for its engine in 1988 [36]. The principal idea of Page Rank is to make use of the link structure to simplify the Web

search. Page Rank evaluate the quality (importance) of the pages by using the following formula:

$$P(A) = \frac{1-d}{N} + d\sum_{i=1}^{n} \frac{P(t_i)}{C(T_i)} \tag{12}$$

Where $P(X)$ is the Page Rank for page X, and $C(X)$is the number of outgoing links for page X. n is the total number of pages, T_i is the i^{th} link computed, and d is a damping factor. This equation can be explained as follows: The page rank value of a page increases the more the page is referenced (linked) by other pages with a high Page Rank value, and the Page Rank value of a page is distributed equally between the outgoing links. Another way to see the page rank, it is to consider a surfer who starting into a random page will jump to another employing one of the links present in the page. This idea, should also include the probability that the user jumps to a random page not linked by the actual page to avoid being stuck in disconnected sub-nets and pages with no links (dangling nodes). The page rank is calculated offline because the high computational power required. The process is as follow:

It starts by generating an adjacency matrix, where all the dangling nodes are grouped together. A dangling node represent a Web page with no links, and it is represented in the matrix as a o row.

Then, the matrix is subdivided in three parts, dangling nodes, nodes related to the dangling nodes (weak non dangling nodes) and the strong non dangling nodes.

The dangling nodes can be lumped into a single node and its page rank computed separately, something similar happens with the weak non dangling nodes.

- For the remaining nodes a Page Rank vector or an approximation is calculated with diverse techniques depending on the implementation [36, 43, 14, 19, 20] of (Eq. 12), which is used to order the result pages for a query.

While this approach certainly has important properties like the rising quality of the results for a given query, the reduction of the manual work and time to search the Web millions of pages, it is not without its problems like giving higher ranks to universally popular pages than to relevant pages and not taking into account relevant pages that do not include the query words [20]. This is way different approaches have considered important topics like page authority [20], and semantic similarity [10]. They were developed because in a rough way Page Rank assign a value to the pages based in their popularity, but a

popular page is not always relevant to the query topic. Further complication arises when relevant pages have low density of the query words.

A solution can be implemented by establishing hubs as a sets of pages that contribution to the rank of an authority page, thus eliminating pages that even though popular are not relevant [20]. On the other hand, semantic similarity [10] seeks to establish meaning relationships between the pages, and it uses the relationships between the pages to improve the Web search by seeking the query meaning and not the query text.

4.1.2.2 *A Web Structural Analysis Application: Influence Detection in Social Networks*

The work "Influence Maximization for Big Data Through Entropy Ranking and Min-Cut" [41] seek to find the most influential nodes by employing the network linking structure. It defines the Social Network as a graph $G = \langle V, E \rangle$, where each user is a node $v \in V$, and every link $e \in E$ represents an influence between nodes. Links are considered unidirectional meaning that given one node has influence into another the opposite is not necessarily true. Each link is formally defined as $E \subseteq \{(i, j) \in V \times V : i \neq j\}$. This is congruent with the network structure in Twitter, where user links represent followers.

Here, a weight Matrix W is employed to represent the influence degree between the nodes. W is mathematically defined as

$$W = \{n\} \times \{n\} \rightarrow N \tag{13}$$

where $n = |V|$, W_{ij} means link weight from node i to node j and $i \neq j$. In Twitter the nodes represent users and the links followers relationships, but it should be considered that these relationships may not represent influence. For example, even if user a follows user b, if user b does not write any tweet, there is no influence of user b in the user's opinion. Furthermore, this work converts the graph into an influence graph, where nodes with a positive opinion belong to the support group, mathematically expressed as $\psi_i \in N^+$, and nodes with a negative opinion are grouped in the opposition set,$\psi_i \in N^-$

Although many ranking methods use node degree related metrics [36, 43, 20], this work considers that node's degree could not accurately represent importance and decides to implement an heuristic based in graph entropy. Graph entropy can be defined by the following equation:

$$H(G, P) = \min_{x \in StableSet(G)} \sum_{v \in V}^{n} p_v log(p_v) \tag{14}$$

The previous equations requires to find an Stable Set (SS) of nodes.

Definition 10. This set is defined as the node collection L such for every member there is no edge to another member.

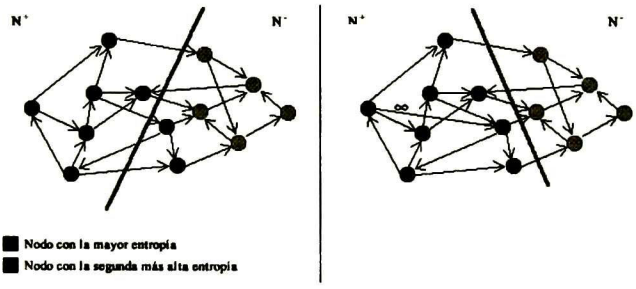
It is known that the SS problem is an NP-Complete problem, so to overcome the problem this work adopts a three step heuristic [45]:

1. Compute node entropy $E(v)$.
2. Compute graph entropy with out node v,$EN(v)$.
3. Calculate the effect with the following functions

$$H(G,P) = \sum_{v \in V}^{n} p_v log(p_v)$$
$$effect(v) = EN(v)/log(EN(v)/E(v))$$

These steps are applied for $\forall v \in V$ to calculate all the effects, then the nodes are sorted by their ranking/effect. With this process entropy could be calculated for different levels where each level is related with direct or indirect influences between the nodes. In the work was determined that with one and two levels in depth the balance between precision and time was reached.

After having obtained the previous data the combination of min-cut and greedy methods are applied to cluster the nodes. The Min-Cut split the graph in two setsN^+ and N^-, for this it is needed to establish source and sink nodes. Then each node $i \in N^+$ is labeled positive and each node $j \in N^-$is labeled negative. Afterward the m nodes with the highest entropy ranking are selected to obtain a min cut.



This process seek a set of nodes which will attract the most nodes when converted.

Figure 1: Finding the most influential nodes in a network graph

This min-cut is employed by a greedy strategy to find the vertices which will attract the most nodes if converted. The algorithms for both the Min-Cut variant and the Greedy Strategy are given next as their appear in the original work:

Algorithm 4.1 Ranking Algorithm

Require: Directed Weight Graph G.

$\Psi'^{+} = \emptyset$
$N^{-} = mincut(G)$
Combine all N^{+} nodes as a super-node
$entropy_{N^-} = H(N^{-}, P_{N^-})$
for all v_i^{-} in N^{-} **do**
  $G^{-} = N^{-}/v_i^{-}$
  $entropy_{G^-} = H(G^{-}, P_{G^-})$
  $effect[i] = entropy_{G^-}/log(entropy_{G^-}/E(v_i^{-}))$
end for
Sort(effect)
for j=1 to $|N^{-}|$ **do**
  if $v_j^{-} \notin \Psi^{+}$ **then**
    $\Psi'^{+} = \Psi'^{+} \cup \{v_j^{-}\}$
  end if
end for
Ψ'^{+}

Algorithm 4.2 Cut Algorithm EMin

Require: Extra seed set Ψ'^{+}

$j = 1$
for $i = 1$ to m **do**
  $\Psi^{+} = \Psi'^{+}[j]$
  while $\Psi^{+} \in N^{+}$ **do**
    $j = j + 1$
    $\Psi^{+} = \Psi'^{+}[j]$
  end while
  $W_{s\Psi^-} = \infty$
  $N+ = mincut(G)$
  Combine all N^{+} nodes as a super-node
end for

Algorithm 4.3 Cut Algorithm with Greedy EMinG

Require: Extra seed set Ψ'^{+}

```
1: j = 1
2: for i = 1 to m do
3:   for j = 1 to m + m' do
4:     a = mincut(N+ ∪ Ψ'+[j])
5:     if a > max then
6:       max = a
7:       ψ+ = Ψ'+[j]
8:     end if
9:   end for
10:  W_sψ- = ∞
11:  Combine all N+ nodes as a super-node
12: end for
13:
```

4.1.2.3 *A Web Structural Analysis Application : A Phishing Site Detection*

With the expansion of economic activities in the Web, a new problem has appeared, the proliferation of replicated sites that seek information for illegal appropriation [27, 26, 37, 46]. With this in mind, developing tools to detect dangerous sites has become a necessity. Examples of these tools are the phishing detection algorithms for suspicious pages based in the Web graph, which allow to characterize legitimate pages. This page characterization varies from a purely manual blacklisting to the compounded weighting of selected features. Examples of these characteristics are the visual themes, page's text, domain names, Google assigned Page Rank, age of domain, suspicious characters in the url, and the appearance of known images.

Once a set of characteristics is defined, a vector for each page is created, where each value in the vector corresponds to one of the characteristics. Then, it is possible to select a criteria to find the suspect pages. This criteria is most of the time the application of a threshold to the dot product of the metrics values [27, 26, 37, 46]. After evaluating each page with the given criteria, a set of suspicious pages is obtained. Here, depending the implementation, one or more actions could be taken:

Adding the suspicious pages for manual revision.

Blacklisting such pages.

Giving a feedback to the system.

Applying posterior analysis of the results.

Thus, even if the concept is simple, creating a good characterization of the pages such that it does not relies in human supervision is not

a simple matter. Most of the actual approaches require human intervention to select the real phishing pages from possible false positives [27, 26, 37, 46].

4.2 SENTIMENT & EMOTION IDENTIFICATION

In this section, several algorithms proposed to try to identify and classify the emotional content in a text are explored to illuminate this new nascent area of research [32, 30, 33, 31, 38], where the range of analyzed text can go from brief text fragments as mails to literary works. The review of these algorithms is essential because the idea of representing the Web data based in its emotional content is the main idea behind this work.

4.2.1 *The Emotion Lexicon: A Different Kind of Dictionary.*

Words can be associated with emotions or sentiments, as an example both delightful and yummy express joy while cry indicate sadness. In the work "NRC Emotion Lexicon" [29], a new kind of dictionary was developed where instead of storing the word meaning, it focuses on the emotional and sentimental associations of each word. This lexicon annotates words with the eight basic emotions proposed by Plutchik [38].

The word-emotion association lexicon was created by using three independent systems:

- Amazon's Mechanical Turk [8]. An online crowd-sourcing platform used to obtain human annotation on the linguistics tasks.
- Roget's Thesaurus [15]. A public domain source of words definitions.

  Google $n-gram$ corpus [25]. It is used to select the word list to be annotated.

The process is as follows, first, the word list is created by selecting only words with an occurrence of more than 120,000 times in the Google $n-gram$ corpus. Then, the Roget's Thesaurus categories for each word is obtained where if a word is ambiguous then it appears in more than one category. With this annotations at word-sense, different level are obtained. Next, the annotation gathering process is started:

- A word choice question is automatically generated for the target. In this step, a word with a high occurrence is selected and the user is asked to identify the closest match for the word. One option includes a near synonym from the thesaurus, and the other options include randomly chosen words as distractions. This question gives the annotator the desired sense of the word.

- If the annotator answers correctly the previous question then he is asked if the target is associated with a positive sentiment or negative sentiment and which emotions expresses.
- If the word choice question is wrong then the annotation data for the associated emotions is discarded.

Each term was annotated by five different persons, in the 74.4% of the instances the five annotators agreed on the term association with a particular emotion. For the rest, 16.9% of the times four of five people agreed, the information for a particular term is obtained by a majority vote. The lexicon includes entries for about 24,200 word-sense pairs. The information for different word senses is obtained by the combination of the associated emotions to each word sense. The previous process resulted in the word-level emotion association of 14,200 word types.

The Emotion Lexicon stores relationships between the words and the emotional categories.

Entries in the NRC Emotion Lexicon version 0.92.

aback	positive:0	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0
abacus	positive:0	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:1
abandon	positive:0	negative:1	anger:0	anticipation:0	disgust:0	fear:1	joy:0	sadness:1	surprise:0	trust:0
abandoned	positive:0	negative:1	anger:1	anticipation:0	disgust:0	fear:1	joy:0	sadness:1	surprise:0	trust:0
abandonment	positive:0	negative:1	anger:1	anticipation:0	disgust:0	fear:1	joy:0	sadness:1	surprise:1	trust:0
abate	positive:0	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0
abatement	positive:0	negative:0	anger:0	anticipation:0	disgust:0	fear:0	joy:0	sadness:0	surprise:0	trust:0

Figure 2: Example of the NRC Emotion Lexicon Entries[32]

4.2.2 *E-mail Analysis: Emotion & Sentiment Tracking*

This section examines the paper "Tracking Sentiment in Mail: How Genders Differ on Emotional Axes" [33] for the basics on sentiment analysis in small texts. The paper propose two ideas for the e-mail analysis:

- Analyzing the data by the writer gender.
- Analyzing the data by the e-mail category.

For this, three distinct e-mail categories were selected: Love letters, suicide e-mails and hate mails. The base of the gender analysis is that men and women will use language differently [5]. For both cases, the analysis seeks to visualize the differences in the emotions expressed by the e-mail in distinct categories. For this, the Relative Salience (RS) metric was defined:

$$RS(w|T_1, T_2) = \frac{f_1}{N_1} - \frac{f_2}{N_2}, \tag{15}$$

where f_1, f_2 are the frequency of appearance of the word w, and N_1, N_2 are the total number of word tokens in the texts. Thus, while

a simple ratio function may be effective in representing the extent to which one emotion is more prominent than another in the text, the RS measure helps to also visualize the source of these emotions. This is because, the metric permits a comparison of each emotion word contribution in both texts.

The experiments to validate the RS measure were performed over a compilation of 348 postings in lovingyou.com, 279 pieces of hate mail sent to Millennium Project, and 21 suicide notes taken from Art Kleiner's Web site [1]. For the gender analysis the Enron e-mail corpus [2] was selected as the source of work-place communications because it consists of more than 200,000 e-mails sent between October 1998 and June 2002 by 150 people in managerial positions. The gender based data set was pruned by removing all mails with fewer than 50 words or more than 200. In addition, emails were removed when they were sent or received from persons which name was not a clear indicator of gender.

Once the data set were formed a text analysis is done by calculating the ratio of the number of words associated with each emotion over the total number of emotion words in the text. This approach is useful to determine if a certain text has more emotional expressions compared to others, even if it is not reliable in concluding if a particular sentence expresses an specific emotion. Then, using the formula in (Eq. 15) between different e-mails, the paper provides a brief analysis and concluded that there was a notable difference in the emotion expressed in each email category. For example, the study shows that women use and receive a larger number of joy and sadness words while men prefer trust or fear words. Additionally, it was found that in cross-gender communication anticipation emotion words spike. For the e-mail categories was observed that there is more fear emotion words in suicide notes, disgust and anger is predominant in hate mail, and love letters include more joy. This results were consistent with the expectations and the previous research.

4.2.3 *Literary Works: Emotion & Sentiment Identification*

When researching the human emotions, the principal problem is the great ambiguity in the terms employed. What constitute an emotion, a feeling or a sentiment basically changes from author to author [32]. Many works and classifications have been proposed in the division of the basic principal emotions, and between them is Plutchik's research [38]. His work conforms the base for the emotion categorization used by this thesis. Nevertheless, it is not the purpose of this thesis to add to the area, but to employ the resources provided by previous works for the data mining of the emotion content in the Web.

With this in mind, we proceed to define the terms of emotion, feeling and sentiment in the way that will be employed by this thesis.

First, emotions can be seen as categories where joy, sadness and anger are example of such categories. Then, feeling can be understood as the actual physical representation of an individual to an emotion. Thus, while each person shows anger in different and particular ways, the emotion is still the same. The final term "sentiment" is more related to the particular person's opinions, and many works [32, 30, 33, 31] represent the sentiment in a positive, neutral or negative expression.

This thesis intends to perceive the Web community stance to a query topic by establishing the general sentiment to the query, and analyzing the emotional output of the pages. In other words the Web stance is closely related to the sentiments and emotions expressed.

By analyzing the Web text, we can obtain a value for the sentiment tendency of a page by counting the positive, and negative words present in the page. In a similar way, the emotion content of a page can be expressed by counting the words of each emotion category [29, 32].

Given the continual change in words connotations, and the effect that these words have in society, several proposals have been given to create emotional lexicons by surveying the target population [32]. For example, works for selecting a classification of base sentiments have been made by researchers in psychology [4, 17, 18, 31, 38, 40].

Base emotions are those which, by varying their intensity and mixture, can create an emotion range in which almost all words can be classified. The classification used in this work associates words with the following sentiments: Joy, sadness, anger, fear, trust, disgust, surprise, and anticipation [38].

In order to help in the sentiment analysis, the concept of emotion word density is introduced.

Definition 11. The emotion word density, $\rho(C_i, F)$, can be defined as the quantity of words associated to a sentiment, $C_i(F)$, for certain number of words present in a document or text fragment, w_f.

$$\rho(C_i, F) = \frac{C_i(F)}{w_f} \tag{16}$$

The emotion word density was employed in the literary analysis of texts like novels, tragedies and tales. By employing the word density as a metric in Web analysis it is also possible to recognize the polarity of the word, which can be positive, negative, or neutral. The polarity of a word is useful to expresses the bias of the word towards an entity, where the entity can represent a topic, a subject or an object of interest. Then, in order to calculate the word density for an emotion category, the rate of related words for every text fragment is measured. In the analysis of literary works made by Mohamed et al [32] a text fragment is composed by 10,000 words. Finally, the analysis results showed a comparison in the internal emotion flows in books from the same time period or topic, and the correlation of

social events and the increment of the emotional expression on those books [30].

4.3 CLUSTERING ALGORITHMIC

This section detail the base for the cluster algorithm employed in the proposal implementation: The Quasi Clique Merger Algorithm (QCM). The QCM is an algorithm that surges from the ideas of Graph Theory and groups the nodes using the idea of pseudo cliques.

4.3.1 *Quasi-Clique Merger Method (QCM)*

This algorithm focuses on employing two ideas to build the clusters: First, the density of a sub-graph, and second, the contribution of a new node to the sub-graph which where defined in (Ch. 3). This is accomplished by using the emotion word density and the emotion salience metrics to group the pages with similar emotional and sentimental content. Also the QCM algorithm permits a multi-membership of the nodes, which can help to visualize subcategories in the clusters. For example, sales pages which include or not product reviews. This method is based in the ideas taken from the graph theory area, specially the cliques also approached in (Ch. 3). Thus, in a weighted graph, where each edge is associated with a weight $w(e)$, the idea of a clique is not properly defined. Then, it is necessary to define the concept of a Quasi-Clique. In order to do this, first, it is necessary to define the sub-graph density and the vertex contribution metrics.

Definition 12. The density of a sub-graph represents the input of each edge in the sub-graph between the combinatorial number of vertices. This metric is defined by the following equation:

$$d(C) = \frac{2\sum_{e \epsilon E(C)} w(e)}{(|V(C)|)(|V(C)|-1)}, \tag{17}$$

where $V(C)$ is the set of vertices in Clique C.

After this, it is possible to define the vertex contribution metrics.

Definition 1.10

The vertex contribution metric is defined as the contribution of the weights of each edge from the vertex v to a vertex u in the Quasi Clique, $u \epsilon V(C)$. This metric is given by the formula:

$$c(v, C) = \frac{\sum_{u \epsilon V(C)} w(u, v)}{|V(C)|}. \tag{18}$$

Given the previous metrics it is possible to define a $\Delta-$Quasi Clique as any sub-graph whose density is greater than an positive real num-

ber Δ. In other words a sub-graph C is a Δ−Quasi Clique of graph G if $d(C) \geq \Delta$.

Now, a basic example is presented to better understand the density and contribution concepts. For this, the following graph (Fig. (3)) represents the linking configuration of a basic sub-net: There is eight Web sites in the sub-net and the edges weights represent the number of links between the individual sites.

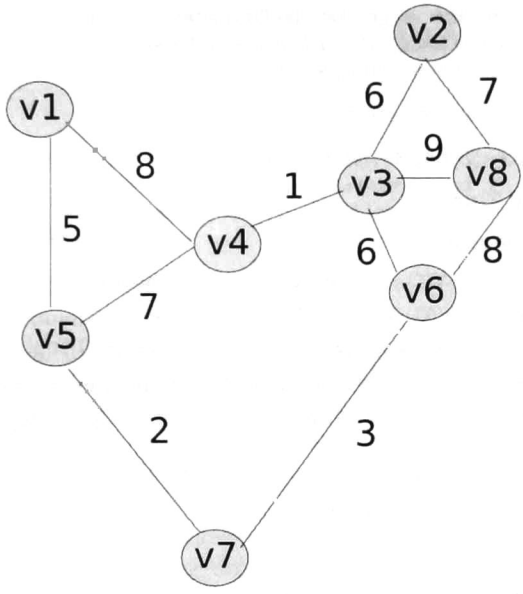


Figure 3: Web Graph Example

In the previous case, we consider two quasi cliques $C_1 = v_1, v_5$ and $C_2 = v_2, v_3, v_6, v_8$. Then, we will obtain the density of C_1 and the contribution of v_4 to C_2 as follow:

$$d(C_1) = \frac{2\sum_{e \epsilon E(C)} w(e)}{(|V(C)|)(|V(C)|-1)} = \frac{2(20)}{3(2)} = 6.66$$

$$c(v_4, C_2) = \frac{\sum_{u \epsilon V(C)} w(u,v)}{|V(C)|} = \frac{1}{4}$$

Using these concepts, the original QCM algorithm [35] can be explained as follows:

The Quasi Clique method add a node to a Clique when its contribution surpasses the Clique density

Algorithm 4.4 QCM Algorithm

Step 0.

$\ell \leftarrow 1$ where ℓ is the indicator of the levels in the hierarchical system.

$w_0 \leftarrow \gamma \cdot \text{rmax}\{w(e):\forall e \epsilon E(G)\}$ where $\gamma(0<\gamma<1)$ is a user specified parameter.

Step 1. (Initial Step).

Sort the edge set $\{e\epsilon E(G):w(e)\geq w_0\}$ as a sequence $S=e_1,\ldots,e_m$ such that $w(e_1)\geq w(e_2)\geq\ldots\geq w(e_m)$. $\mu \leftarrow 1, p \leftarrow 0$, and $L_\rho \leftarrow \emptyset$.

Step 2. (Starting Search).

$\rho \leftarrow \rho+1, C_\rho \leftarrow (e_\mu)$. $L_\ell \leftarrow L_\ell \cup C_\rho$

Step 3. (Grow).

Substep 3.1

If $V(G)-V(C_\rho)=\emptyset$, then go to step 4 otherwise:

Pick $v\epsilon V(G)-V(C_\rho)$ such that $c(v,C_\rho)$ is a maximum.

If $c(v,C_\rho)\geq \alpha_n d(C_\rho)$ where $n=|V(C_\rho)|$ and $\alpha_n=1-\frac{1}{2\alpha(n+t)}$ with $\alpha \geq 1$ and $t\geq 1$ as user specified parameters, then $C_\rho \leftarrow C_\rho \cup v$ and repeat step 3.1

Substep 3.2

$\mu \leftarrow \mu+1$. If $\mu>m$ go to step 4.

Substep 3.3

Suppose $e_\mu=xy$. If at least one of $x, y \notin \bigcup_{i=1}^{p-1} V(C_i)$ then go to step 2, other wise go to substep 3.2

Step 4. (Merge).

Substep 4.1

List all members of L_ℓ as a sequence $C_1,\ldots,C_s$ such that $|V(C_1)|\geq|V(C_2)|\geq\ldots\geq|V(C_s)|$ where $s\leftarrow|L_\ell|$.

Substep 4.2

If $|C_j \cap C_h|>\beta\min(|C_j|,|C_h|)$ where $\beta(0<\beta<1)$ is an user specified parameter, then $C_{s+1} \leftarrow C_j \cup C_h$ and the sequence L_ℓ is rearranged as follows:

$C_1,\ldots,C_{s-1} \leftarrow$deleting C_j, C_h from $C_1,\ldots,C_{s+1}$ and $s\leftarrow s-1, h\leftarrow maxh-2,1$.

Go to substep 4.4

Substep 4.3

$j\leftarrow j+1$. If $j<h$ go to substep 4.2

Substep 4.4

$h\leftarrow h+1$ and $j\leftarrow 1$. if $h\leq s$ go to substep 4.2

Algorithm 4.5 QCM Algoritm (Continuation)

Step 5

Contract each $C_\rho \epsilon L_\ell$ as a vertex:

$V(G) \leftarrow [V(G) - \bigcup_{p=1}^{s} V(C_\rho)] \bigcup \{C_1,\ldots,C_s\}$,

$w(uv) \leftarrow w(C_{i'},C_{i''}) = \frac{\sum_{e \epsilon E_{C_{i'},C_{i''}}} w(e)}{|E_{C_{i'},C_{i''}}|}$

If the vertex u is obtained by contracting $C_{i'}$, and v is obtained by the contraction of $C_{i''}$ where $E_{C_{i'},C_{i''}}$ is the set of crossing edges which is defined as $E_{C_{i'},C_{i''}} = xy$: $x \in C_{i'}$, $y \in C_{i''}$, $x \neq y$. For $t \in V(G) - C_1,\ldots,C_s$, define $w(t,C_{i'}) = w(t,C_{i'})$. Other cases are defined similarly.

Step 6

$\rho \leftarrow \rho+1$, $L_\rho \leftarrow \emptyset$, $w_0 \leftarrow \gamma \max w(e): \forall e \in E(G)$ where $\gamma (0<\gamma<1)$ is an user specified parameter and go to step 1 (to start a new search in a higher level of the hierarchical system).

END.

The QCM algorithm was the first clustering approach, the idea of multi-membership and the evaluation of belonging based in the contribution to a cluster greatly called into us. The QCM running time bound of $\mathcal{O}(n^2 log n^2)$ was considered acceptable and an implementation was made, unfortunately the system required evaluation of all the relationships between the nodes in a cluster and the next candidate which made the program unfeasible. It is still a great influence in our concept of clustering even if no longer literally applicable in its majority but in its conceptual ideas.

Part III

THE PROPOSAL

"I keep the subject constantly before me, and wait 'till the first dawnings open slowly, by little and little, into a full and clear light."— Sir Isaac Newton [34]

5

SYSTEM DEFINITION

In this section, a description of the proposed system is given. Such system is built using the following modules:

1. The data gathering module.
    a) The crawler.
    b) The page selector.
    c) The Web page text acquirer.
2. The data processing module.
3. The clustering module.

Each one of the modules will be described in the following sections.

5.1 DATA GATHERING MODULE

This module includes a wide variety of tasks, from gathering the linking information of the subnet to acquiring the text of each Web page. Thus, it was necesary the creation of sub-modules to better control this process. In this section each one of this sub-modules are briefly aproached, also it should be noted that the system can operate over any Web page data set with an html format. This module was only employed to gather the data for the analysis and can be modified at the user convinience.

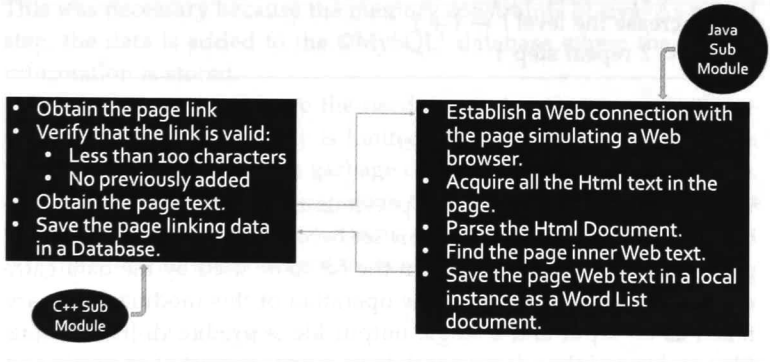


Figure 4: Data Gathering Process

5.1.1 *Crawler Module.*

The initial module is the most simple on the system. It seeks the linking information of the sub-net. Such information is distributed in the following way:

P_0={initial link set of the 100 most relevant pages, given by Google}.

- P_1={all links of each page in P_0}.

P_2={all links of each page in P_1}.

As it is seen, this work relays in the data obtained by a search engine like Google. While the search engine data is the base, our work still needs a functional crawler module to expand into the page selection. The theory of how to create a full crawler system is given by Gupta et al. [12]. However, because the limitations imposed by bandwidth and time, a smaller and simpler implementation was done for this work. Thus, the principal crawler subsystem can be defined by the following code:

Algorithm 5.1 Crawler Algorithm

Step 0
Initialize the initial data set $S_0 \leftarrow [SearchEnginePages]$
Initialize the current level $l \leftarrow 0$
Initialize the first level link set $S_{l+1} \leftarrow \emptyset$
Step 1
For each page p in the current level l
Obtain all links in the page p
For each link o
if $o \rightarrow length < 100$
$S_{l+1} \leftarrow S_{l+1} \cup [o]$
Increase the level $l \leftarrow l + 1$
If $l < 2$ repeat step 1

5.1.2 *Page Set Formation Module.*

With the data obtained by the previous module an edge set can be built for the Web sub-graph. This set becomes essential to obtain the page list which will the conform the set to be used by the data gathering module. Therefore, for the operation of this module 3 files are taken as an input and a single output file is produced. In the input files are located the edges in each level, a edge consist of an origin and a destiny node. Here, each node is represented by a page url except for those contained in the first file whose origin node is represented by the "root" value.

The collection of the pages can be done by the following steps:

1. Taking a link for analysis.
2. Determining if the current link is a valid page url.
3. Determining if the current link has been added to the page set.
4. Adding the link to the set and repeating the process for each link in the input files.

Even, if these steps are easy to understand, for sets with millions of links they can become rather stressful to implement. Given the constrains in time and resources, only a limited selection system was implemented for the testing of the main ideas.

Now, it is necessary to explain the second and third steps, in the previous process. These steps determine if a link is a valid page and if it has been added previously. In order to determine if a link is a Web page, a validation on the size of the link is made to avoid spurious links. Clearly, this is not the best approach, thus future work shall provide more constrains like verifying the link structure and composition. For the third step, it was necessary to implementation of a binary tree where the links are stored letter by letter, given the random letter appearance in the links the binary tree could be formed and processed at most lineal time with respect the number of links.

5.1.3 *Web Page Text Acquirer.*

The data gathering module takes the obtained non-redundant links, and it gets the info of each page. This information refers to the Web Text present at each page, but it can be expanded to contain also the html distribution, and the visual style.

For this module, it was needed to implement a simple C++ application that call a Java application (jar) to perform the page parsing. This was necessary because the memory constraints of java. As a final step, the data is added to the ©MySQL[1] database where the linking information is stored.

It is important to observe the need to employ C++ as an intermediary because Java memory is limited to 2GB, and it is not possible to gain direct control of the garbage collector. These limitations work against the needs of any application that analyzes hundreds of thousands pages. In contrast C++ has no parsing utilities or classes, thus it is difficult and inconvenient to create parsing functions on it.

5.2 DATA PROCESSING.

Once the data has been gathered, its necessary to format it. By this, we mean to express the data in a way that can be easily employed by the clustering methods. For this, we can see the process as:

1 http://www.oracle.com/us/products/mysql/overview/index.html

To convert the page text in a weighted word list.

- To convert this list into emotion vectors.

For the first steep, we employ a B-Tree, where each node has:

27 links (one for each letter).

- An associated character given by the father link.

  A string that represents the word which is created by adding its predecessors characters.

  A weight to represent the times that the word appears in the page.

Also we have a division of nodes as follows:

The root node:

- It has " " as the associated string.
- All its links point to the null node when initialized.
- It has a "0" weight.
- It has no correspondence to any word in any page.

The null node:

- It has " " as the associated string.
- All its links point to the null node when initialized.
- It has o weight.
- It has no correspondence to any word in any page.

The new node:

- It has an associated character given by the ASCII representation of $c = (l_p + 60)$ where l_p is the father linking position.
- It has an associated string which is the result of appending the father associated string and the associated character.
- All its links point to the null node when initialized.
- It has a weight equal to the times that the word appears in the page.

Once the B-Tree has been built, an in-order exploration is made to create the Emotion Vectors. An emotion lexicon is necessary at this step, and such lexicon was given by Mohammed et al. [29]. It is possible to see that an emotion lexicon is an especial type of dictionary, which instead of having word definitions, it stores the word emotion associations.

In this case, the lexicon is formed by an association of words and emotion categories. Such categories are joy, sadness, anger, anticipation, disgust, fear, surprise and trust [38]. In addition, the sentiment categories (positive and negative) are included [33, 31, 32]. In the lexicon the association of a word to a category is represented by 1, and a word can be associated to multiple categories.

In order to explore the B-Tree, an emotion vector is created for the page by:

- Initializing the page vector as an array which will contain a component for each one of the emotional or sentimental categories.

  Detecting the lexicon words present in the page. While each word could be considered to have their own set of emotional and sentimental relations, it is not possible at the current time to generate a mapping of the emotional input of each word of each language. The previous problem have lead to employing the emotional lexicon by Mohammed et al. [32, 29] to evaluate the sentimental and emotional expression of each page.

  Adding the weight of each word node to all the associated categories represented in the array. This means that for each lexicon word present in the page the word emotional input is added to the vector. This will produce a vector which will store the word number of relations to each emotion category.

As a final output this module generates a vector list. Each row in the list correspond to a Web page where each element stores the emotional and sentimental data. Also for each page the url, and an identifier is included.

5.3 CLUSTERING.

With the data formatted into a vector form to represent each page, the clustering process can be initiated. The clustering module receives as input a page vector list, in which each line represents the data from the page. The format of a typical line in the list ca be seen in (Fig. 5):

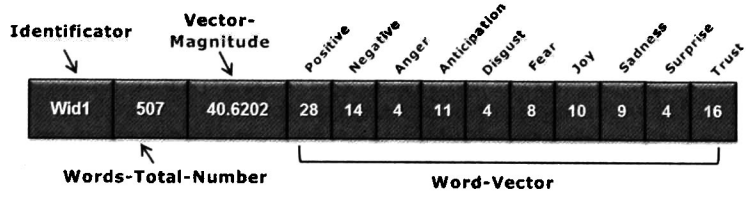


Figure 5: Line format of the Vector List

Representing the Web pages as a vector permits to employ clustering algorithms for the analysis.

As can be seen in (Fig. 5), the list lines contain a word vector where each component represents the number of words present in the page which are related to the corresponding sentimental or emotional category. With the previous data a new vector is obtained for each entry: The percentile-vector that stores in its components the fraction of the page text related to each emotion or sentiment category. Once these vectors, one for each page, are obtained, the clustering process can be started by obtaining two parameters: The variation and the separation factors. In the next sections, the description of both factors is explained.

5.3.1 *Variation Factor*

Given two percentile page vectors $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$, the variation between their components is measured by the variation factor. In order to define the variation factor, first it is necessary to create a new vector, $\overrightarrow{a} = \overrightarrow{V_1} - \overrightarrow{V_2}$. Thus, the components of this vector will be formed by the differences between the components of the initial vectors.

Thus, the definition of the variation factor can be given by the following formula:

$$variationFactor(\overrightarrow{V_1}, \overrightarrow{V_2}) = ||\overrightarrow{a}|| = \sqrt{a_1^2 + a_2^2 + ... + a_n^2} \quad (19)$$

It is important to note that this metric would be equivalent to obtaining the euclidean distance between the peak points of the initial vectors, which is given by the formula:

$$d(\vec{p}, \vec{q}) = d(\vec{q}, \vec{p}) = \sqrt{(q_1 - p_1)^2 + \ldots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (20)$$

Once this factor is obtained, it can be seen as the distance between the two initial vectors: The greatest the distance, the further apart they are. This factor can be described as a way to measure how different are the emotional and sentimental expressions in two pages.

5.3.2 *Separation Factor*

This factor takes into account three aspects in order to measure the distance between the two page vectors:

The variation factor between the vectors.

The differences in the standard deviation of the vector components.

The Chebyshev distance [9] between the vectors.

In a similar form to the variation factor, this aspects are used to conform a new vector $\overrightarrow{b}$. Each vector component will store the value of an aspect and the separation factor will be obtained by the next formula:

$$separationFactor(\overrightarrow{V_1}, \overrightarrow{V_2}) = \frac{||\overrightarrow{b}||}{||\overrightarrow{r}||}, \tag{21}$$

where $\overrightarrow{r}$ is the cluster representative given by the average vector.

This can be used to reflect how different are two pages in their emotional expression by taking into account the maximum difference of emotion expression and the standard deviation of their components. In layman terms this could be visualized as taking into account the maximum separation in the emotions present in the pages and the differences how much each sentiment or emotion varies respect to a mean.

5.3.3 *Clustering the Pages*

With the separation factor, the next step in the clustering algorithm can be explained. This is done by taking the vectors in the page vector list and obtaining communities where the individual pages have a great emotional or sentimental similarity. For this process, this work implements an algorithm based in the QCM Algorithm [35]. All the theory about this algorithm is explained in (Sec. 3.3) .

Once the data is represented as a graph, the QCM algorithm can be used to produce the page communities where the sentimental and emotional expression is the most similar. While the QCM algorithm has an acceptable time processing boundary of $\Theta(n^2 log n^2)$, the time required to represent the data as a complete graph in large data sets can be unmanageable. Thus, in this work a modification of the process has been made in a way that the ideas of cluster density and the contribution of a new vector to a cluster are taken into account, and they are implemented in the following clustering algorithm.

Algorithm 5.2 Clustering Algorithm

Step 0
Define $i \leftarrow 0$
Define $cList \leftarrow \emptyset$
Define $pSet \leftarrow [allPageVectors]$
Step 1
For each $pVec \in pSet$:
If $cList = \emptyset$
Add a new Cluster c, $cList \leftarrow cList \cup [pVec]$
Define the cluster representative $\overrightarrow{r} \leftarrow pVec$
Else
For each Cluster $c_i \in cList$
Define the cluster representative $v \leftarrow (c_i \rightarrow \overrightarrow{r})$
Define the contribution of the vector $pVec$ to the Cluster c_i
$c(\overrightarrow{v}, C) \leftarrow 1 - separationFactor(\overrightarrow{v}, \overrightarrow{r})$.
If $c(\overrightarrow{v}, C) \geq \alpha$
Add the vector $pVec$ to Cluster c_i, $c_i \leftarrow c_i \cup [pVec]$
Actualize the cluster representative, $\overrightarrow{r} \leftarrow \frac{1}{N}\sum^{N} r_i$
If the vector $pVec$ was not added to any cluster in $cList$
$cList \leftarrow cList \cup [pVec]$

With the previous algorithm the clustering can be made without requiring computing all the edge weights in the graph. For example, the computation of all the edge weights in a data set of more than 500,000 page vectors, using the QCM algorithm, took more than a week before crashing, and the edge list document could not be opened with any text processor in windows or Linux. Those problems forced the implementation team to adapt the algorithm. Thus, instead of evaluating all of the edge weights, the idea of a centroid was implemented. This centroid represents the average vector of the cluster vertices. Also given that most graphic software can not correctly represent multi-membership, this characteristic was suppressed. Then, by only obtaining the separation distance between the page vector and the current cluster representative, a functional program was created which can organize the page set in less than a 1000 seconds.

Our current algorithm takes the idea of multiple metrics for clustering with the separation and variation factors. The algorithm described in this section divides the set into clusters where each page is sorted into a single cluster. This form of grouping does not permit multi-membership in contrast with the QCM algorithm. It should be taken into account that many of the graphical tools for representing graphs have problems with multi-membership.

The complexity of this algorithm is $O(pn)$, where n is the number of clusters and p the number of pages. Given the algorithm, if $n = 1$, all the pages are in a single cluster which reduces the complexity of the proposed algorithm to $O(n)$. In the other hand, when $p = n$

then each Web page is in a singular cluster which gives the algorithm a complexity of $O(n^2)$. In layman terms, this algorithm starts with a vector list as an input, and for each page explores if it should be added to an existing cluster, if there is not an apropiate cluster the system creates a new cluster which contains only that Web page.

6

VALIDATION & VERIFICATION

Given the initial page set size, around 630,000 pages, the number of clusters formed by each of the test runs were numbered in the thousands. It is not the purpose of this work to give a detailed account of each cluster formed, but to give a summation of the ten principal clusters obtained in each run such that the structure of the data could be inferred. Thus, it is necessary to explain how the data was obtained. An exemplification of the gathering process is as follow:

- Making the query in the Google search engine.

  Obtaining the 100 most relevant results from Google.

  Making an in depth two level search for all links in the pages.

- Obtaining all the non repeated links with urls less than 100 characters in length.

In this chapter the results of three test runs are detailed by first given a brief summation of the process, and then displaying the results obtained to end with an interpretation for the data acquired.

6.1 ANALYZING ALL THE EMOTIONS: THE FULL VECTOR APPROACH

Once the process was defined, the first approach was clear: To submit the full page vectors and process all the current data. The idea behind this process was to find communities that displayed similar emotional expression patterns. Then, the analysis of the communities could be used to label the different page types by using their emotional expressions. In the analysis of the data set, two test runs were employed to evaluate the system.

The distinction between the this tests is the acceptable range of divergence between the candidate page and the current cluster representative. This value is set as an external parameter α where $0.01 \leq \alpha \leq 1$, which represent the maximum percentage of divergence for a page to belong into a cluster. The divergence value is given by the separation factor detailed in the previous chapter (Chap. 5.3). In the first run, the alpha value was set to 0.07, indicating that at most a candidate page could have a divergence of 7% respect the cluster representative to belong in the cluster. In the second run the alpha value was set at 0.2.

6.1.1 *The First Test Run:* $\alpha = 0.07$

When the Alpha value was set at 0.07 the vector list was divided into 302,455 clusters in a time of 25,376.61 seconds, for a initial set composed of 629,096 page vectors. By setting $\alpha = 0.07$, only the pages which deviate at most 7% respect the current cluster representative will be grouped together. For the obtained clusters only the cluster analysis for the ten with the most pages is included, and in (Fig. 6) can be observed the clusters representation.

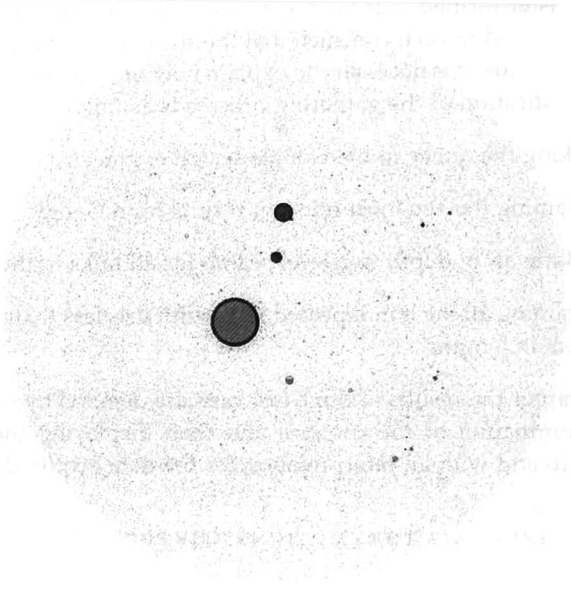


Figure 6: Pages in each cluster

The clusters with more pages are represented by bigger spheres. Most pages are not included in big clusters because the restrictions in similitude.

Given the space constrains of this work in the next image is resumed the emotional input of each cluster by detailing the emotion percent of each category (Table 1).

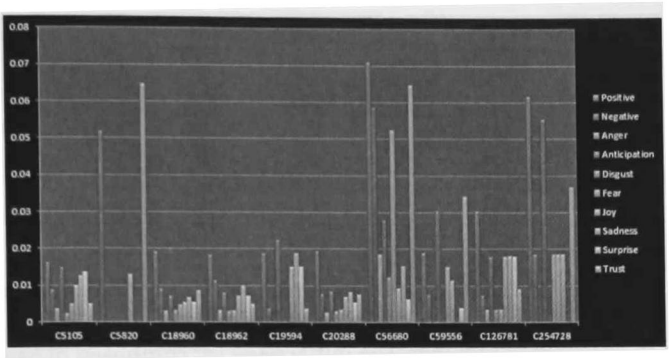


Table 1: Emotion Expression by Cluster

In the previous graphic, a cluster is represented by a set of lines where each line indicate the mean expression level of an emotion. From the obtained results, the following pattern in the standard deviation between the vector components was observed (Table 2).

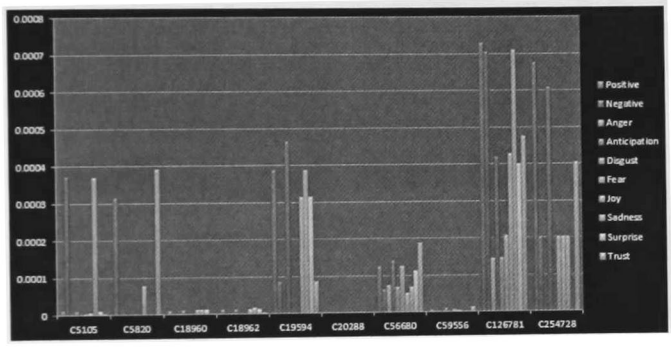


Table 2: Cluster Components Standard Deviation

In (Table 2), the emotion categories of each cluster are represented by a color bar. The clusters appear organized by the number of pages contained, and it is visible that the highest inner emotion variance in a cluster is around a 0.075% respect the cluster representative. Also, the previous graph show that some strong emotion variations are curved by the small variation in other emotional categories. For example in the first cluster the variation of both the positive and negative sentiment is around 2.3%, which is stabilized by the small variation in the disgust emotion category around 0.08%.

The pages in the big clusters are so similar that the process could be use it to find plagiarism works and phishing pages.

The previous data can be used to reach some interesting conclusions, the first one by noting the large number of generated clusters. A possible cause of this is that the system is only grouping together pages which display a similar emotional expression in all of their categories. With an alpha set at 0.07 the maximum separation distance

between the cluster's mean and a prospect vector to be added should be no greater than 7%. Thus, this process would create clusters with a high enough similitude that its output could be used to find duplicated pages and pages with strong resemblance in their content as is the case in plagiarized works.

6.1.2 *The Second Test Run:* $\alpha = 0.20$

In this run the alpha value was set at 0.2, which would permit pages within a 20% emotional/sentimental divergence from the cluster centroid to be grouped together. The initial page list composed of 629,096 page vectors was divided into 80,008 clusters in 3,715.037 seconds. As in the previous case in this section are only detailed the data of the 10 clusters with most pages. In (Fig. 7) is given the clusters representation for this run.

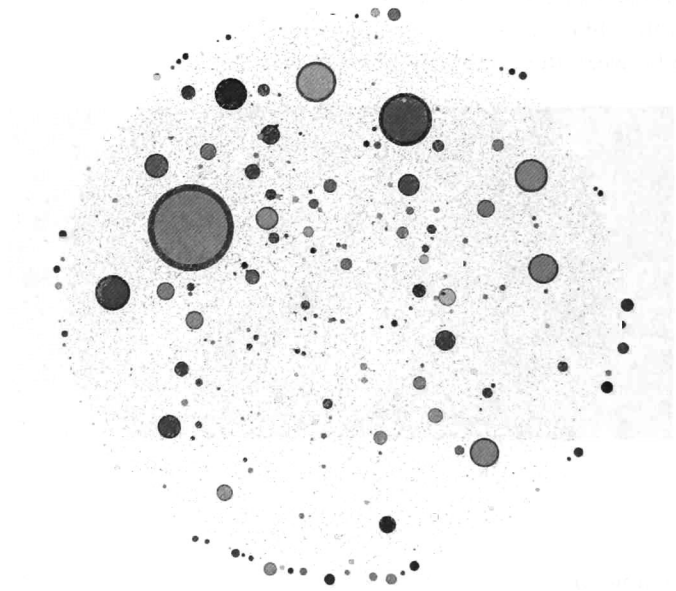


Figure 7: Pages in each cluster

A drastic reduction in the number of clusters means the pages are not so similar.

Given the space constrains of this work, the next (Table 3) is presented to resume the emotional input of each cluster by detailing the emotion percent in each category.

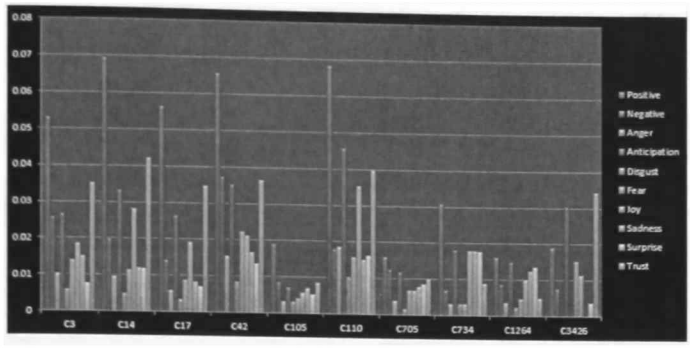


Table 3: Emotion Expression by Cluster

In (Table 3), each color line represent an emotion category while the clusters are expressed in the x-axis. Also, the comparison of the inner emotional variance of the vector components of each cluster is given in (Table 4).

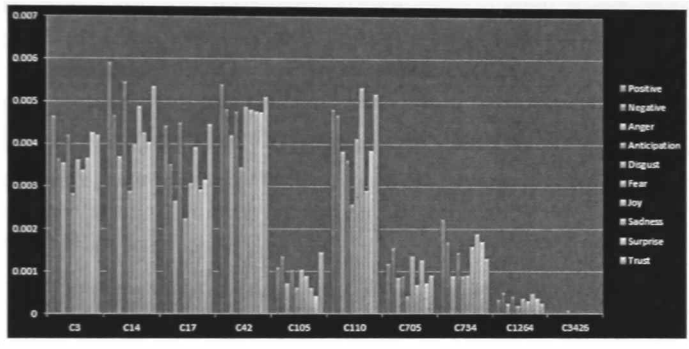


Table 4: Cluster Components Standard Deviation

In (Table 4), the emotional internal variation of each cluster is represented by a color bar for each category. The clusters appear organized by the number of pages in them, and it is visible that the highest expression variation is around a 0.06% which is a magnitude order less than the average vectors. It is also shown that in some clusters the inner emotion variance is relatively high for each category, while in others it is barely present at all. In this case, no inner variance in the categories may present the idea that some of the clusters contain Web pages with the same content but different urls.

The internal standard deviation in the clusters are still an order of magnitude less than the clusters averages.

6.2 ANALYZING VECTOR FRAGMENTS

While the previous process gives a lot of data, it was considered that many applications only require the sentimental input for the analysis.

In addition, many applications only require to analyze one or two emotions instead of the full spectrum. Thus, in order to improve the efficiency of the proposed system, it could be better to modify the system's parameters to account for only the relevant categories . In this section, the experiments were repeated over the same data while only taking into account a limited set of active vector components. As a result, an active vector component is defined as the one that stores the number of words related to an emotional or sentimental category of interest for a specific user.

6.2.1 *Sentiment Analysis: Positive-Negative*

This was the first repeated test for the active component system. In this test, only the components for Positive and Negative sentiments were taken into account. The system was programed for operating with an alpha of 0.07, which will translate into grouping together pages with at most a 7% separation factor from the cluster mean .

The process grouped 629,096 page vectors in 235.053 seconds into 2003 different clusters. In this section is reported only the results for the cluster with the largest number of pages. It is possible to compare this analysis with similar data with different active components, this comparison is detailed in the conclusion area of this chapter. In order to begin the analysis, a graphical representation of the clustering (Fig. 8) is highly useful.

Figure 8: Cluster Representation

In (Fig. 8) each cluster is represented by a color sphere, the sphere size is representative of the data contained in each cluster.After this, the (Table 5) is given to represent the mean positive and negative sentiment present in each cluster.

The clusters formed by only taking into account the sentimental data take a lot less time for the same page data set.

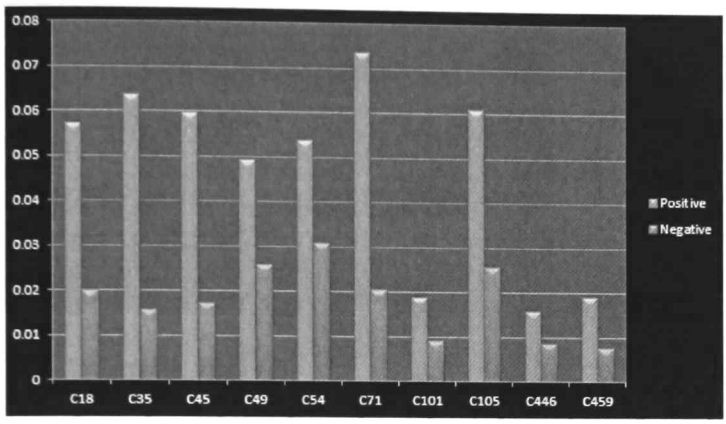


Table 5: Positive-Negative Cluster Averages

By analyzing (Table 5) is easy to see the difference in the sentimental expression between the principal clusters. While this can be used to compare the clusters by themselves, it is also important to observe the inner variance in each cluster. In order to observe this variance, (Table 6) is given.

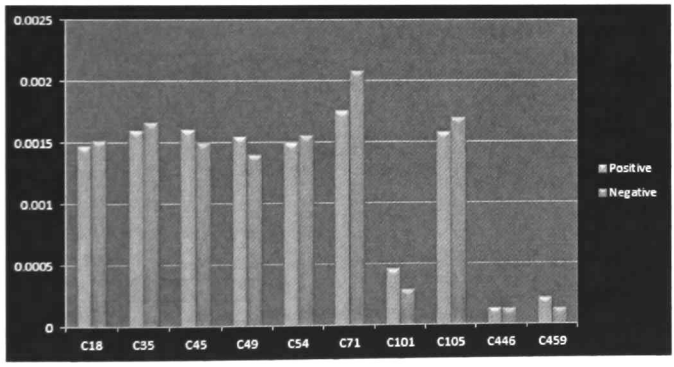


Table 6: Cluster Components Standard Deviation

The internal standard deviation in the clusters components are more alike.

(Table 6) shows the internal sentimental variance in the pages in each of the principal clusters. The inner variance can be seen as a metric of how much the pages sentimental components differ respect to the cluster representative. This is possible because the cluster representative is defined as the mean vector.

6.2.2 *Emotion Analysis: Joy-Anticipation*

In the previous case, only the sentimental categories were taken as an input for the analysis process. Here, the idea of selecting only some emotional categories is explored. Thus, the joy and anticipation categories were selected as the active components.

By setting alpha in 0.07, it was expected a similar process output and running time as in the previous case, with perhaps a difference in the cluster formation. This was the case as the process conformed 2204 clusters with a running time of 224.009 seconds over the same input data.

Given the number of clusters obtained as an output, the results shown in this section correspond to the data obtained for the ten clusters with the largest number of pages. This data is compared in the (Sec: 6.3) with the results from the other tests. In order to begin, the (Fig. 9) gives the reader a graphical representation of the clusters obtained.

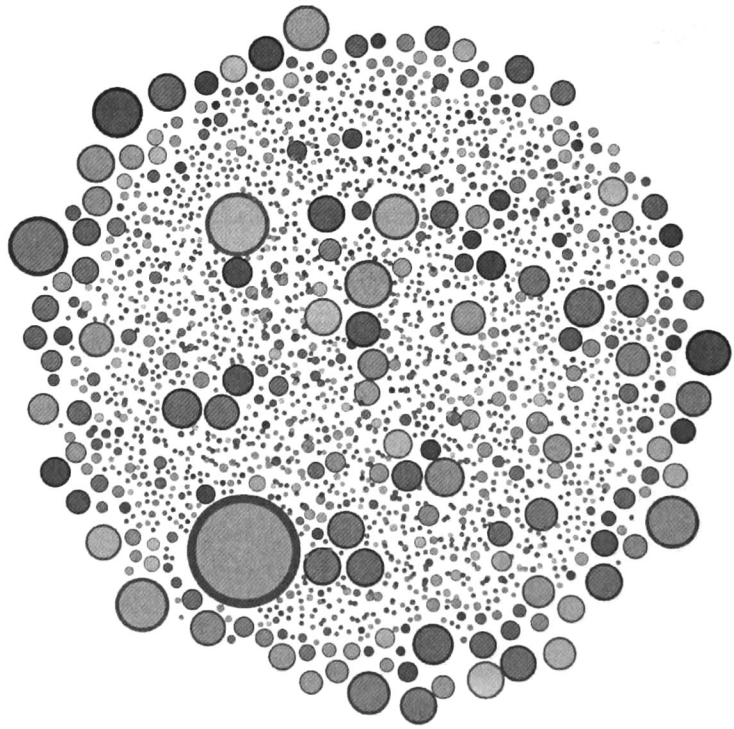


Figure 9: Clusters Representation

In a similar way to the previous case, in (Fig. 9) each sphere represents a cluster where the sphere size is given by the number of pages

in each cluster. After this, the (Fig. 7) provides the mean values for the selected emotion categories in each cluster.

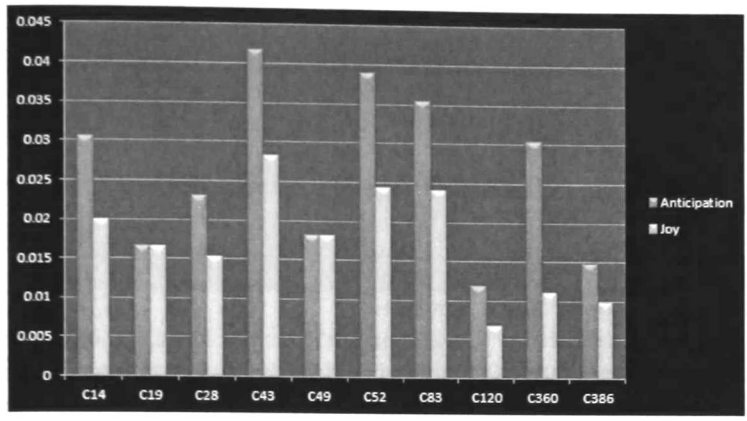


Table 7: Joy-Anticipation Cluster Averages

The (Table 7) graph express the average percent of text related to the joy and anticipation categories. While this can be used to compare the clusters by themselves, it is also important to observe the inner variance in each cluster. Thus, (Table 8) is provided.

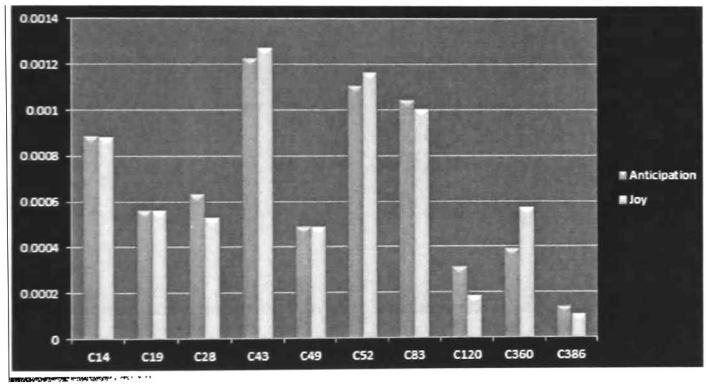


Table 8: Joy and Anticipation Cluster Deviation

(Table 8) shows the inner emotional variation in each one of the principal clusters. The inner variation can be seen as a metric of how much the pages sentimental components differ respect the cluster representative. This is possible because the cluster representative is defined as the mean vector.

6.2.3 *Emotion Analysis: Anticipation*

As a final test, the idea of employing the system with only one active category is done. In this instance, the anticipation category was selected as the active component in the vectors. The setting of alpha in 0.07 and the employment of the same input data was made for comparison purposes.

In this test, the process obtained only 130 different clusters in an operating time of 164.369 seconds. This is clearly an order of magnitude less than the previous tests, and it is rationalized by taking into account that here, the system only makes a hierarchical aggregation of the pages by their level of expressed emotion. Thus, 130 levels are enough to classify all the input data.

The data detailed in this section corresponds to only the first 10 clusters with the largest number of pages. Also, this data is compared with the results of the previous test in the (Sec. 6.3). In order to being, lets start with (Fig. 10) in which the clusters obtained are given a graphical representation.

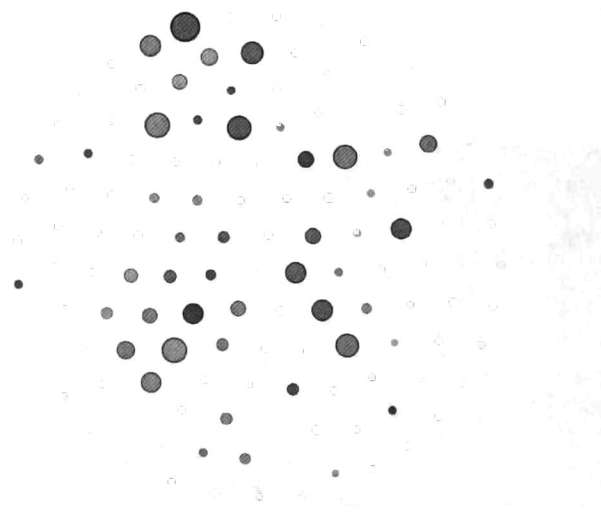


Figure 10: Clusters Representation

After giving the clusters representation in (Fig. 10), the next step is to provide the reader with the mean values of the anticipation emotion in each cluster, (Table 9).

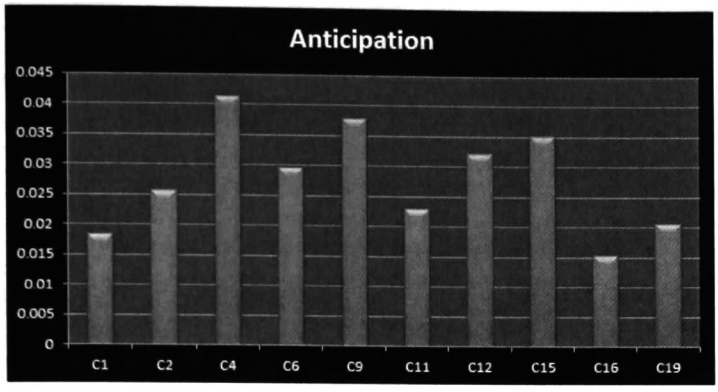


Table 9: Joy-Anticipation Cluster Averages

Once this is done, the inner variance of the clusters should be taken into account (Table 10). The inner variance in the top ten clusters in this case is an order of magnitude less than the expressed mean values. This is important because it assures the user to the correct conformation of the clusters.

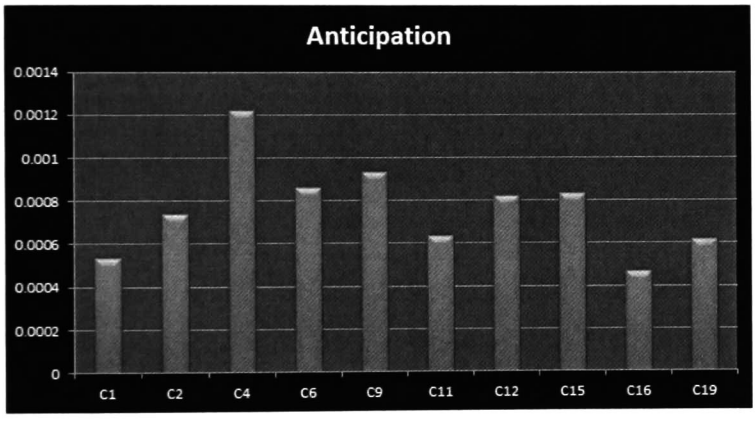


Table 10: Anticipation Cluster Deviation

After concluding this test, in the following section (Sec. 6.3), it is given an analysis of the results. This comparison is done by analyzing the differences in the test inputs, outputs and behavior. Also, the number of clusters, the running time and the number of active components are take into account. As a final step, the analysis provides suggestions in which this application could help to resolve some of the problems today encountered in the Web analysis area.

6.3 INFERENCES FROM THE EXPERIMENTS

This section starts by giving the general inference about the system. The current system has two principal operating modes:

1. Analyzing all the emotion and sentiment categories present in the vector list.

2. Selecting the active categories to analyze.

The previous aspect give the system the capability to adapt depending the necessities of the user. In the first case, by taking all the emotion and sentiment categories into account, the pages are grouped together only if they contain a similar emotional expression in all the categories. While the second approach can be personalized to focus in certain emotional or sentimental categories.

Thus, while for example, the first clustering mode could acquire aggregations of extremely similar pages, in the second mode the user would define the clusters based only in the sentimental content of the pages.

The principal application of this system is to find Web page aggregations of interest for the user. This could be done in any number of ways depending of the user, in here the acquisition of a data set was performed by crawling the related pages to a query. Other methods could include analyzing a whole site in which the system could identify the pages with the highest emotional concentration. Also, analyzing the Web for encountering pages with a emotional firm could be done to find racist sites, phishing pages or suicide groups.

Given the time limitations and the laboratory resources, only a single data set could be obtained for this thesis and further exploration of these topics should be done in future work. For this implementation, it was discovered that an alpha value between 5% and 20% yield the best results.

Next, the results obtained in each one of the test are compared. Thus, the reader is given a description of the test run, the observed behavior and then the analysis conclusion is denoted. Also, a comparison of the number of clusters obtained, the running time and the number of active components in each test is expressed in (Table 11), (Table 12) and (Table 13).

Lets start with the tests of the first mode, by setting the alpha value at 0.07 and 0.2 respectively an appreciable change in the performance was obtained. In the first case, by setting the alpha value at 0.07 the system could only group together pages which were so close in all their components as to be the same which generated a lot of singular page clusters. Thus, the running time was exponentially increased because the system evaluates a page against all the possible clusters before making a new one. In contrast, the second test generated only

80,000 clusters with a reduction in the running time of an order of magnitude.

In both tests the clusters inner variance was within an acceptable threshold, but in the first one many of the clusters contained only repeated pages. Therefore, the author considers that an alpha value of 0.2 for the first mode is an acceptable setting.

For all the tests of the second mode, the alpha value was set at 0.07 which was considered the most adequate value given the reduction in the analyzed vector components. Here were explored some combinations of emotional and sentimental categories.

In the first test, the analysis of the data was conducted with only the sentimental categories active, in the second joy and anticipation were selected, and finally in the third test only a single category was taken, anticipation.

The results obtained show a similar performance in the outputs of the tests with the same number of active vector components. This reinforce the idea of a correlation between the number of categories and the clusters generated. The results for the last test are an order of magnitude less than those of the ones with two categories which also conforms with the previous expectations.

Now, the (Table 11), (Table 12) and (Table 13) are provided to grant the reader a graphical representation of the results obtained.

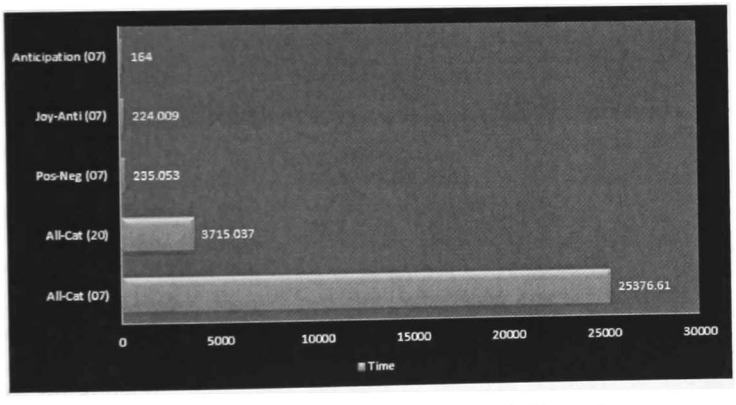


Table 11: Running Time Comparisons in Seconds

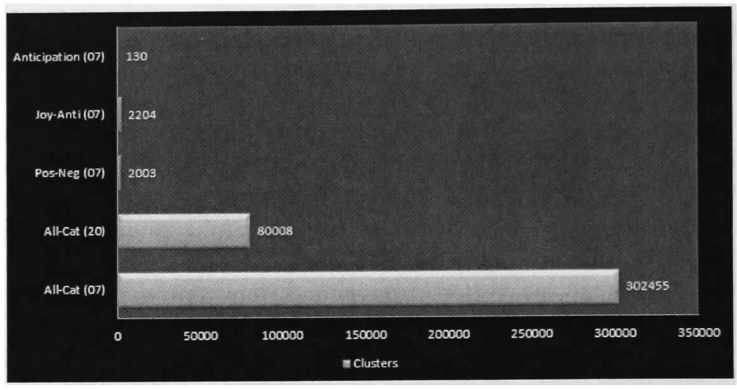


Table 12: Clusters Generated by Each Mode

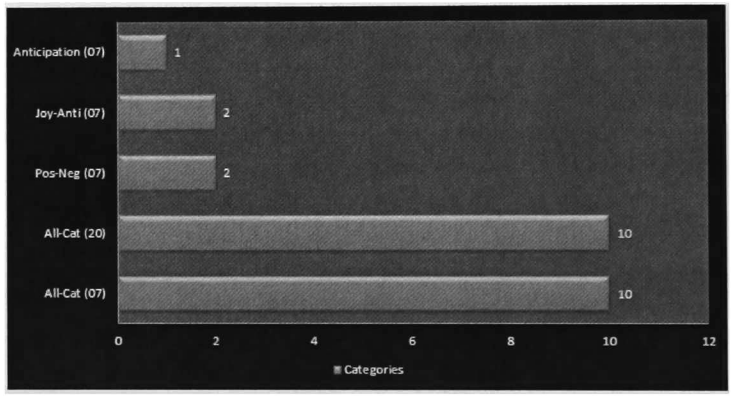


Table 13: Active Categories in Each Mode

All the test runs were made over the same input page set which it is constituted by 629,096 page vectors. The previous figures show a marked correlation between the processing time and the number of clusters generated. This can be explained by looking at the algorithm complexity of $\mathcal{O}(np)$ where n is the page number and p the number of clusters, because of this the algorithm depends on the number of clusters as much as in the number of Web pages. Thus, the user should consider the alpha value carefully to obtain the minimum number of possible clusters reducing the processing time.

6.4 DERIVED WORKS

6.4.1 *Sentiment & Emotion Analysis for e-Services*

Using the previous implementations, it is possible to propose several improvements over the current search engines. For this, a new system, with the capacity to categorized the results based in the page type and

its emotional and sentimental content, is proposed. For example, the tests realized show that around 40%-60% of the principal results for a first query correspond to pages with little interest to the user. In order to improve this, the proposed system can be adapted to classify results by taking into account the emotional and sentimental content of each page. Making possible to split the page set, being queried, into the supporter and opponent classes.

In addition, by also taking some structural page proprieties, it is possible to refine the selection by page type. In a second idea, the proposed system can be used to compare similar products by acquiring the sentimental response to each product in the the product review and forum pages. This system can give the user a comprehensive report to facilitate the decision making process. Finally, the measurement of the emotional response in certain categories can be used to identify some risk inducing Web communities such as hate groups and their support sites.

While this systems have not been fully implemented, ideas are begin studied to find their feasibility. In the following part, each idea is described and a possible proposal for their implementation is added.

6.4.1.1 *The Categorization of The Search Engine Results*

Here, the previous ideas of classifying the page set by their support level and the division by page type are explored. Part of this research was presented by Santos et al.[42], in the conference paper "Sentiment Analysis for E-Services"

In the paper, a system to classify the principal results to a query made in Google was defined. This system took the emotional and sentimental data expressed in the page vectors and conformed a rough set of communities with similar emotional expressions.

To do this, the system employed an early version of the proposed system in this thesis. This implementation worked by taking all the emotional and sentimental categories in consideration but only considering the variation factor for the clustering process.

The results of this configuration was a considerable increase in the inner variance of the components of the vectors in each cluster. The detailed configuration and results can be perused in the annexed paper.

For the second idea, it was postulated that by adding structural characteristics [27] to the page vectors a division by page type was possible. Some of the proposed characteristics are:

The visual structure of the page.

- The page height and width.
- The total word count.
- The page title.

- The page domain.
- The presence of user reviews and comments.

While this characteristics are normally employed in the identification of phishing sites [27], it is the author opinion that they also can be used to complement the emotional and sentimental information of a Web page to produce an acceptable typification on the input set.

The previous statement is based in the results obtained in the test of the paper system. For example, when querying about a product most of the pages would be sales pages. The emotion expression in the sales pages is greatly impacted by the user reviews but the system would conform the clusters by only seeing the emotional data. This generated clusters which included pages of different types. By recognizing the page type by its structure, the system could correctly classify each page into the support or opposition classes corresponding to its type.

The previous concepts were implemented initially in Beta version as part of the proposed system in the paper to categorize the search engine results. However, due to lack of time and the scope of this research, the finalization of the system was included in the future works.

7

CONCLUSION

The work done for this thesis has achieved the principal objectives in proposing and implementing a system with the capacity to cluster an heterogeneous set of Web pages. This system has been tested with an input set of almost 630,000 pages, in diverse configurations to analyze the emotional and sentimental data present in the Web. Some of the ideas of this system were presented by Santos et al. [42] in the conference paper "Sentiment Analysis for e-Services"

The current system operates by clustering the Web pages into communities based in the resemblance of their emotional and sentimental data. In order to do this, the system employ the following modules to gather, process and finaly cluster the Web page data:

1. The data gathering module.
    a) The crawler.
    b) The page selector.
    c) The Web page text acquirer.
2. The data processing module.
3. The clustering module.

In the first module the text of the input Web pages are acquired to be used as the input of the following modules. In order to realize this task, the system queried a search engine and obtained all the relevant Web pages to the query, (Sec: 5.1.1) and (Sec: 5.1.2). Also, the text of each Web page was represented by a word list which would conform the input to the next module, (Sec: 5.1.3). The second module, described in (Sec: 5.2), would transform the word list of each page into an emotion/sentiment page vector, and it would also create a vector list with all the non-cero vectors. At last, the clustering module, (Sec: 5.3), will group the Web pages based in the emotional and sentimental expression of the active categories.

The previous system permits the user to personalyze it to his convinience. For example, by selecting all the pages of a Web site, the user could then find the set of pages with the most negative expressions. Also, the active vector categories permit the user to only focus in either the sentiment expression, or any category set of interest which could greatly benefit the user. Therefore, it is the author opinion that this system could be employed in the following tasks with minor adequations:

Typifying the Web pages categories.

Finding communities or interest groups:

- Detecting support groups.
- Detecting opposition groups.
- Detecting suicide risk pages.
- Detecting hate sites.

The base of this system is the research of Mohammad et al. [29, 32, 30, 33, 31], who proposed employing the emotional and sentimental data to analyze literary works [30] and typefy e-email categories [33, 31]. The proposed system in this thesis goes an step further by analysing the data of an heterogeneous set of Web pages, where in most ocassions the pages include texts of many authors. Furthermore, it expands the analysis to pages where no-standard exist on how they organize their information by the use of clustering techniques to avoid get trampled by those organizational issues.

As a final author note, this system should not be taken as a final version but the steping stone to a new generation of specialized systems that account for the emotional and sentimental data in the Web. This belief surges from the idea that not only the humans need to transmit their emotions in anything the do, but also need to identify themselves with the emotions of another which conforms the base of our society. Therefore, ignoring the emotional and sentimental output in the Web is an error that should be corrected.

Part IV

APPENDIX

Sentiment Analysis for e-Services

F. Santos-Sanchez, *Member, IEEE* and A. Mendez-Vazquez, *Member, IEEE*

***Abstract*—New e-services come on-line each year at an exponential rate. Most of them have the need to analyze and interpret enormous quantities of data. However, many of them do not take into account the emotions and sentiments in the Web page for their analysis. Thus, in this work, we proposed a novel system to obtain data of interest from a Web search engine by analyzing the emotional and sentimental content of each page. This information can provide a new way to identify the community response to a product: Given the positive vs negative average sentimental rates in the forums, it is possible to know the Web community perception to the product. In addition, this classification makes possible to identify the best sales options given a certain product, and it also permits to obtain a great reduction in the needed time to find the best options by grouping the comparison pages for an easy access . Thus, we believe that this system is the beginning of a new way to help the user in their selection of the best product to buy, and it makes possible to have a better understanding on how the users perceive the recommended products.**

***Index Terms*—e-service, emotion, sentiment, analysis, clustering algorithms.**

I. Introduction

WITH the exponential Internet expansion in recent years, many new opportunities, problems and resources have appeared. Identifying, analyzing and understanding such aspects are the base of the e-services [1]. One e-service that can be used in a great range of applications is the detection, recognition and analysis of opinions [2], [3], [4]. This is of great importance for areas like sociology, politics and marketing [5], [6], [7].

In this work we seek to implement a new idea: The Web community opinion about a product can be obtained by analyzing the sentimental and emotional text at the principal Web pages given by a search engine. This pages will be clustered by their sentimental ranks to form diverse communities (like sales pages, bug reports, Wikis, and products reviews) in which the analysis of their emotional content will be used to establish a Web opinion about the query. The analysis of sentimental and emotional text has been previously used in the study of literary works by Mohammad et al. [2], [8], [3], [4].

Now, the idea of finding Web communities is not a new one, and it has been employed in the detection of phishing Web sites [9], hate groups [5] and Web piracy domains [10], [11] to say some examples. However, we have never seen an attempt to find communities based on feeling analysis, thus the novelty of this proposal.

A referral e-service can be seen as the union of three systems: the categorizing of the Web pages given by the search engine, the identification of the communities and the emotional analysis of said communities to form the a general opinion of the product. This paper focus on the first system by detailing a novel system of Web page categorization through feeling analysis. First, in section II a general description is done about document analysis, Web analysis and community detection. Then, in section III, a description of of the proposed work is given from the crawler and data gathering modules to the cluster module based on feelings. In section IV, a series of experiments about client satisfaction based on a series of laptop models are run to test the proposed system. Finally, at section V, we discuss future possibilities for these type of systems.

Fernando Santos is with the Department of Electrical Engineering and Computer Sciences, Cinvestav, Guadalajara, Mexico. e-mail: fsantos@gdl.cinvestav.mx .

Dr. Andres Mendez is with the Department of Electrical Engineering and Computer Sciences, Cinvestav, Guadalajara, Mexico, e-mail: amendez@gdl.cinvestav.mx.

II. Previous Work

In the past years, many works have been released in both document analysis and Web analysis. For both categories, there exist many possible variants, but we will concentrate our review in the most important ones in document analysis and Web analysis.

A. Document Analysis

Document analysis focuses on the aspects related to the gathering, processing and interpretation of the text documents. The problems being researched in this particular area go from plagiarism detection [9], [10] and creation of document signatures [11] to sentiment analysis of literary works [8] and file classification [10]. For example, given a document, Mohammad et al. [8], [2], [3], [4] proposes a method to automatically classify the literary text based in the analysis of the sentimental and emotional words present in the text. In order to do this, the authors needed to create a lexicon [3], [4], which can be viewed as a subtype of dictionary. This dictionary stores the emotional categories of each word at the lexicon. Each element at the lexicon can be seen as a vector, which stores the word and its correspondence to each sentimental and emotional category. The relationship of a word with the ιth emotional or sentimental class, $i = 1, 2, ..., n$ can be represented formally by the indicator function$\delta_w(\iota)$, where $\delta_w(\iota) = 1$ indicates that the word evokes the emotion or sentiment, and $\delta_w(\iota) = 0$ represents the opposite.

It is important to define what is an emotion, what is sentiment and what are their differences. In the work proposed by Mohammad et al. [8] the sentiment content refers to the positive-neutral-negative implication of a word, while the emotional content refers to the emotional category associated with the word.

Base emotions can be seen as those which can have a variation on their intensity, and mixing them can create a

wide emotion range in which almost all words can be classified. Several works [12] for selecting a classification of base emotions have been made by researchers in psychology. For example, Mohammad et al. [8] uses the Plutchik classification [6] which is also used on this work. This classification associates words with the emotion categories of joy, sadness, anger, fear, trust, disgust, surprise, and anticipation. It also includes the positive and negative relationships for each word. A word which is not classified as positive or negative is by default taken as a neutral word, however there are words which can have both a negative or positive connotation. Such words are associated to both the positive and negative sentiments in the lexicon.

To help in the emotional analysis, the concept of emotion word density [8] is introduced. Emotion word density is defined as the quantity of words associated to an emotion that we expect to see every certain number of words. Also the emotion analyzer recognizes the polarity of the word, which can be positive, negative, or neutral, and it expresses the favor of the word towards a subject.

The concepts above explained were applied to the analysis of literature works, specifically novels and fairy tales. By establishing a window of 10,000 words, the study makes a comparison on the respective sentiment density of each of the books. Further more, emotion flows were found in each book, and a correspondence between certain social events and the increments of density in certain emotions was found [8].

B. Web Analysis

We refer as Web Analysis as the set of techniques, methods and tools used to extract and understand the information on the Web. Even if most Web users think that the principal source of information is the Web text, such data it is but just a small percentage of the data contained on the Web. Several works on this area include fields like determining and interpreting the page relationships [13], [14], [15], detection of Web communities [15], analysis of social groups [2], [7], and the development of Web risk detection techniques [5], [9].

These topics have gained significant importance during the last decade, represented by the great number of applications developed to solve these problems. A requirement for these applications is to have a model to represent the relationships on the Web. For example, it is common to represent the Web as a directed graph: Vertex are nodes containing some pertinent information, while the edges between vertex are seen as directed relationships. This model is used in the Page Rank algorithm [14] where vertex are pages and edges represent links.

While the previous graphical proposal is good for human understanding, when employed in the analysis of hundreds of thousands of nodes, it becomes rather impractical. So said information its mostly expressed as the adjacency and incidence matrix representations previously spoken in . After having said this, we continue by briefly describing two cases and compel you to review the referenced works.

1) Phishing Detection: With the expansion of economic activities in the Web, a new problem has raised, the proliferation of replicated sites that seek information for illegal appropriation [9]. With this in mind, developing tools to detect dangerous sites has become a necessity. Examples of these tools are the phishing detection algorithms for suspicious pages based in the Web graph, which allow to characterize legitimate pages. This page characterization varies from a purely manual blacklisting to the compounded weighting of selected features. Some of these characteristics are the visual theme, page text, domains name, Google assigned Page Rank, age of domain, suspicious characters in the URL, and appearance of known images.

Once a set of characteristics is defined, a vector for each page is created, where each value in the vector corresponds to one of the characteristics. Then, it is possible to select a criteria to find the suspect pages. This criteria is most of the time the application of a threshold to the dot product of the metrics values [9], [5]. After evaluating each page with the given criteria, a set of suspicious pages is obtained. Here, depending the implementation, one or more actions could be taken:

Adding the suspicious pages for manual revision.
Blacklisting such pages.
Giving a feedback to the system.
Applying posterior analysis of the results.

Thus, even if the concept is simple, creating a good characterization of the pages such that it does not relies in human supervision is not a simple matter. Most of the actual approaches require human intervention to select the real phishing pages from possible false positives.

2) Community Detection: With the proliferation of the Web services many groups have found an easy access, and low cost media to promote their ideologies. Some of these group ideologies can represent a danger [5], and therefore there is an interest in identifying them and following their actions. This is not the only case where identifying Web communities is required, marketing, social analysis, and search engines development also require a way to solve this problem [5], [9], [13].

In its simplest form, a community is a group of individuals closely interrelated [16]. With each application, the community definition is given by the sought information. For example, Are we searching for terrorist cells in the Web? Or perhaps the adolescent's opinion on a topic in twitter? One approach to form the communities is to start with one community and segregate it into subsets based in the individuals characteristics. Another approach starts with many individuals and cluster the communities based in the individuals similarity. A description of the individuals is necessary in any of these approaches. For this, it is possible to represent the characteristics as vectors. In some cases is necessary preprocess the information to select a subset of characteristics. In those cases the objective is to find a signature for the individuals that includes the minimum number of characteristics and gives an optimal clustering of the individuals.

III. Sentiment Analysis E-Service

In this work, we focus in the current tendency of using the Web to create opinions. In order to this, we propose a

referral e-service. This service takes the principal results for a given query in a search engine and provides the user with the community sentiment and emotion response to the query. The sentimental information evaluates the positive-negative response, while the emotional information gives the user an easy visualization of the evoked emotions by the query.

The service proposed can be visualized as the union of three independent modules. The first module categorizes the pages provided by a search engine, and with most of the user time being spend finding the correct pages, this is quite relevant for anybody. The categorization of the pages is based in two principal factors: The quantification of the emotion content present in the page, and the distribution of such content into sentiment and emotion categories. By this, pages with a high emotion content are grouped together based in their percentage distribution. For example, pages with an emotional/sentimental content of around 40% of the page's text, where the positive sentiment is highly predominant with 30% positive vs. 3% negative of the emotional content, could be grouped together. In a second module, the categories identification is made, in the previous example the analysis shows this category to be Web sellers, where the page is specific to a single product, and it contains no reviews of the product or Q&A areas. At last, in the final module the emotional information expressed in the produced page categories is analyzed, and the Web community response to the product is obtained.

The proposed algorithm for the first system's module can be visualized as a set of specific tasks:

The Data Gathering.
The Data Processing.
The Data Clustering.
The Info Interpretation.
The Info Visualization.

In the following sections, we describe each of these tasks in more detail.

A. Data Gathering Module

The system utilizes the principal results obtained by a Web search engine as an input set. The set is composed by the first n links obtained by the search engine, and once this set is obtained the Web text of each page is acquired. Its important to notice that given the heterogeneous initial set no common HTML structures should be expected. This work takes a $n = 100$ links from Google queries, however, it is important to remarks that such results can differ in function of the location of the queries, the search engine used, and the previous queries made by the user.

The data gathering process is subdivided in two parts:

The acquisition of the linking data:

- The initial set, the 100 first results provided by a query made in a search engine.
- The output links of all the pages in a 2 level depth.

The acquisition of the Web text for each page in the final set.

To obtain all these data a crawler is required. A crawler [17] is a piece of software designed to explore the Web to collect data of interest. The basic procedure of the crawler is to analyze a Web page, find its outgoing links, store the data of interest and explore the links obtained. This work employed a basic crawler designed in C++ for this task.

B. Data Processing Module

This module takes as an input the Web page texts to create weighted word lists as a first step. In a second step, the weighted lists are explored to create the page vectors by using an emotion lexicon. Many studies have defined basic categories for emotions, in particular this work utilizes the lexicon provided by Mohammad et al. [18] which utilize the emotion categories proposed by Plutchik [6] with the inclusion of the positive and negative categories. The page data stored by the module includes the page link, an identifier and the page vector. The ith page vector component is defined by the next formula: $c_i = n_i/t$, in which each component represents the percent of the text related to the emotional/sentimental category c_i, and it is obtained by dividing the number of words related to the category n_i over the total number of words present in the page t.

Once the page vectors have been acquired the information is stored in a formatted file. Such formatting may depend on the clustering method used, and in this case a simple vector list suffices. Some clustering algorithms like the graph based Quasi Clique Merger (QCM) method [19] may require additional processing to create edges between the vertexes. The QCM idea surges from the concept of clique or quasi-clique in the weighted graphs. Furthermore, QCM seeks to group together the pages based in the balance of two metrics: the density of a clique, and the contribution of a new member to the clique. Also, the QCM find cliques and permits multi-membership where an element can belong to one or more clusters. This makes necessary to have a second pass where cliques with many common members are joined.

C. Data Clustering Module

The clustering of the information should take into account at least two aspects: The variation factor and the separation factor. The variation factor can be seen as a measure of the similarity in the emotion expressions at the pages. For example, this will group together pages that have low expression of positive sentiment and a lot of anger. In contrast, the separation factor measures the similarity in the total sentiment and emotion content expressed in the page i.e. this will group together pages by the quantity of emotion content.

In a more formal approach, we propose the next metrics to account for the variation and separation factors:

1) Separation factor:

$s(c_i, p_j) = |\vec{s}|$; $|\vec{s}| = \sqrt{s_1^2 + s_2^2 + ... + s_n^2}$

The separation vector $\vec{s}$ stores the following data in each component:

$s_0 = p_1 - p_2$

$s_1 = n_1 - n_2$

$s_2 = \sigma_1 - \sigma_2$

Where p_1 corresponds to the positive value in the page vector, $n_1$corresponds to the negative value

and $\sigma_1$is the standard deviation in the page vector components. Similar values are calculated for the cluster representative in p_2, n_2, σ_2.

2) Variation factor:
$v(c_i, p_j) = 1 - (eDis(\overline{x}, p_j)/max(|\overline{x}|, |p_j|))$
where $eDis(p_i, p_j)$ describes the euclidean distance,
$eDis(p_i, p_j) = \sqrt{(p_i^1 - p_j^1)^2 + ... + (p_i^n - p_j^n)^2}$.

Both metrics establish the membership of a Web page (represented as a vector) to a cluster. Here, for a page to belong in a cluster, the variation and separation factors of the page with the cluster representative ($\overline{x}$) should be inside a threshold defined by the user. cV is the variation coefficient and basically establish the variance (σ) range in which the prospect pages should be admitted, and also it is user defined.

Many clustering algorithms can be used to obtain the sentiment and emotion communities. However, the final objective is to group the pages proposed by the search engine into communities such that they express a similarity in emotion content and in the emotions expressed. The proposed method for this clustering process is shown next, and it is based in the QCM algorithm [19]:

Sector Clustering Algorithm

```
Initialization
  s ← {p_1, p_2, ..., p_n}.
  C_0 ← s.
  S_0 ← {C_0}.
  sectorArray ← {S_0}.
Loop
  Set o ← |sectorArray|.
  Make S_{o+1} ← ∅.
  Make S_{o+2} ← ∅.
  For each cluster C_i in sector S_j:
    x̄ ← (1/N) Σ_{i=1}^{N} x_ik .
    m ← |S_j|.
    For each page P_k in cluster C_i:
      if s(c_i, p_k) > β
        if v(c_i, p_k) < γ
          if |x| > |p_k|
          S_{k+1} ← S_{k+1} ∪ {p_k}
          else
          S_{k+2} ← S_{k+2} ∪ {p_k}
        else
          C_{m+1} ←
          C_{m+1} ∪ {p_k}.
      if C_{m+1} ≠ ∅ S_n ←
      S_n ∪ C_{m+1}
      if S_{k+1} ≠ ∅ sectorArray →
      sectorArray ∪ S_{k+1}
      if S_{k+2} ≠ ∅ sectorArray →
      sectorArray ∪ S_{k+2}
If j < |sectorArray| repeat the loop.
```

Our current algorithm takes the idea of multiple metrics for clustering with the separation and variation factors. The algorithm described in this section divides the set into clusters and sectors (clusters of clusters) where each page is sorted into a single cluster, and each cluster belongs to a single sector. This form of grouping does not permit multimembership in contrast with the QCM algorithm. It should be taken into account that many of the graphical tools for representing graphs have problems with multimembership.

The complexity of this algorithm is $O(opm)$, where o is the number of sectors (cluster of clusters), p the number of clusters and m the number of pages in a cluster. Given the algorithm, if $m = n$, all the pages are in a single sector so $p = 1$ and $o = 1$ which reduces the complexity of the proposed algorithm to $O(n)$. In the other hand, when $p = n$ and $o = n$ then each cluster can contain only one $m = 1$, which gives the algorithm a complexity of $O(n^2)$.

In layman terms, this algorithm starts with a page vectors set as an input, and creates a single sector with a single cluster that contains the entire set. Once, the initial cluster is obtained the page vectors in the cluster are evaluated. If the emotion content of a page is too distant from the cluster average, then a new sector is generated. If a page has a similar emotion content range then it is left in the sector. A page vector will be put in a new cluster depending on the variation factor between the vector and the current cluster average.

D. Interpreter Module.

This module gets the sectors and clusters arrays, and transform them into tables where sectors represent the level of emotion expression (none, medium, high...), and the clusters represent the different expression groups at each level. For example, high positive, and joy clusters tend to represent pro Web sites, while a high negative sentiment with too much anger could be hate groups. This module is responsible to format the data for a correct visualization, and it depends on the next module.

E. Visualization Module.

The correct visualization of the data can be a complex task depending on the number of pages selected. We propose a graphical interpretation given by Gephi [20], in which case the interpreter module should provide a GDF file, or an executive analysis of the information in the next form:

Graphical representation of positive vs negative sentiment.

- Top 10 more negative pages.
- Top 10 more positive pages.

Average expression of the emotions categories:

- Top 10 joyful pages to the top 10 most hateful pages

Given a large number of pages, it can be convenient to analyze only some sectors so pages with little emotional content do not affect the values. However, indifference feeling can not be correctly interpreted by the current model, and it should be considered a topic of future research.

IV. Results

The tolerance ranges are given by the user defined constants γ and β. However, further research in the correlation of the constants and the query is needed. The empirical analysis in this work has shown that correct value ranges are $0.95 \leq \gamma \leq 0.99$ and $0.95 \leq \beta \leq 0.99$. These ranges generate adequate emotional variations on the categories created by the algorithm.

The initial data set for the pages was obtained by taking the first 50 pages of reviews, and the first 50 pages of problem reports for each query. Then, the process transforms the pages into page vectors by counting the emotional and sentimental words in each page. After deleting all zero vectors, the data is grouped by the previously explained clustering algorithm.

The system was employed to analyze the Web pages data for three Google queries:

Lenovo Ideapad y510p
Compaq Presario CQ40
Asus Q501LA

A. Asus Q501LA

The initial data set was reduced to 81 page vectors. Then, after the application of the clustering algorithm, 14 clusters were obtained. The Asus clusters are shown in the following image:

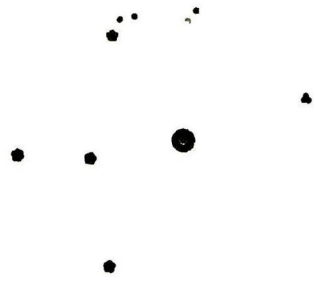


Figure 1: Asus Cluster.

Next an abstract of the principal Asus clusters is given in table IV-A.

Table 1: Asus Principal Page Categories.

Fields	Category I	Category II	Category III
Id	01	02	03
Color	12/215/93	215/12/207	12/93/215
# Members	30	7	6
Average positive sentiment	4.93%	3.05%	2.44%
Average negative sentiment	2.73%	2.04%	4%
Positive standard deviation	0.98%	0.72%	0.49%
Negative standard deviation	0.85%	0.82	0.5%
Peak emotions	Trust	Trust	Sadness
	Anticipation	Anticipation	Fear
	Joy	Joy	Anticipation
Peak emotions avg.	3.06%	2.02%	2.39%
	2.40%	1.53%	2.38%
	1.64%	1.13%	1.80%

These results show that most of the pages have a positive connotation. Thus, trust and anticipation are the most common emotions expressed in the average page. The third category is formed by problem report pages which makes for a greater negative sentiment. It is important to note that the emotions in this category are expressing sadness and fear, instead of joy and trust.

B. Compaq Presario CQ40

The initial data set was reduced to 49 page vectors, after the application of the clustering algorithm 17 clusters were obtained. The Compaq clusters are shown in the following image:

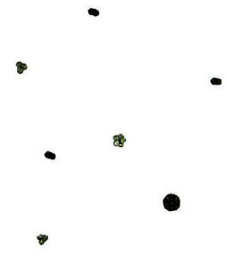


Figure 2: Compaq Clusters.

Next an abstract of the principal Compaq clusters is given in table IV-B.

Table 2: Compaq Principal Page Categories

Fields	Category I	Category II	Category III
Id	01	02	03
Color	51,195,225	215,225,51	225,174,51
# Members	10	6	4
Average positive sentiment	1.49%	3.44%	1.87%
Average negative sentiment	1.35%	1.69%	3.28%
Positive standard deviation	0.81%	0.53%	0.69%
Negative standard deviation	0.74%	0.55%	0.63%
Peak emotions	Trust	Trust	Sadness
	Anticipation	Anticipation	Fear
	Anger	Fear	Anger
Peak emotions avg.	1.37%	3.21%	2.82%
	1.04%	2.32%	2.07%
	0.70%	1.48%	1.31%

The principal clusters for the Compaq show a lower average positive sentiment, which is noticeable because the inclusion of anger, sadness and fear into the emotional expressions in the average page. The low number of page vectors generated corresponds to the decrease in the results obtained by the search engine.

C. Lenovo Ideapad y510p

The initial data set was reduced to 49 page vectors, after the application of the clustering algorithm 17 clusters were obtained. The Lenovo clusters are shown in the following image:

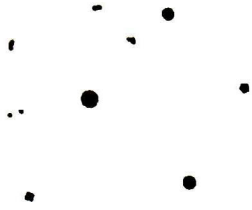


Figure 3: Lenovo Clusters.

Next an abstract of the principal Lenovo clusters is given in table IV-C.

Table 3: Lenovo Principal Page Categories.

Fields	Category I	Category II	Category III
Id	01	02	03
Color	53,23,174	23,38,174	144,23,174
# Members	23	13	11
Average positive sentiment	4.17%	3.44%	5.08%
Average negative sentiment	2.49%	1.69%	1.88%
Positive standard deviation	0.74%	0.53%	0.83%
Negative standard deviation	0.58%	0.55%	0.39%
Peak emotions	Trust	Fear	Trust
	Anticipation	Anticipation	Anticipation
	Joy	Sadness	Anger
Peak emotions avg.	3.13%	0.85%	2.89%
	2.12%	0.77%	2.38%
	1.48%	0.69%	1.23%

The principal clusters for the Lenovo have a great positive sentiment balance, but in the expressed emotions are included fear, sadness and anger. This reflect an increment in the problems related to this equipment.

D. Inference Results

The comparison between the principal clusters is giving the following conclusions:

The emotions contained in the result pages for the Asus model have less negative emotions in average. This means that the community express a favorable emotional response to the product.

Unlike the Compaq and Lenovo models, the peak negative emotions for the Asus principal clusters do not contain anger. This is particularly important because it can be related to a low probability of severe problems.

The previous two points give the Asus model the highest ranking for the products.

The low number of page vectors obtained for the Compaq model is related to the size of the search engine result set. This confers a low popularity for the product given mostly by its negative peak emotions.

V. Conclusions

The current increase in the Web data makes necessary new systems that can analyze and interpret this data to present a compact summary to the final user to help in the decision making process. We believe these novel methodology can be the base in which many of these systems can be personalized to supply solutions to the users in the need of methods of sentiment analysis. Our idea is that the sentimental and emotional expressions used when describing a product or topic can be used to form a perception of the community which can be used to facilitate a decision. We are looking to expand the analytical capabilities of the proposed system to better infer the community expressions to topics as well as products. For example, it is necessary to automate the page segregation into categories (like forums, web sellers, Wikis, index...), and the generation of executive reports that compare two or more similar products to facilitate the decision making process.

Acknowledgment

We would like to especially thank Dr. Mohammad and his collaborative group for their work in the sentimental and emotional analysis,and their commitment to improve the area of computer science by sharing their resources.

References

[1] A. Tiwana and B. Ramesh, "E-services: problems, opportunities, and digital platforms," in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, Jan 2001, pp. 8 pp.–.

[2] *Tracking sentiment in mail: How genders differ on emotional axes*, 2011.

[3] *Emotions Evoked by common words and phrases: Using mechanical turk to create an emotion lexicon*, 2010.

[4] *Generating high-coverage semantic orientation lexicons form overtly marked words and a thesaurus*, 2009.

[5] R. D. E. F. X. Qi, K. Christensen, "A hierarchical algorithm for clustering extremist web pages," *International Conference on Advances in Social Networks Analysis and Mining*, 2010.

[6] R. Plutchik, "Emotion: Theory, search, and experience," *A General Psychoevolutionary theory of emotion*, vol. 1(3):3-33, 1980.

[7] L. L. B. Pang, "Opinion mining and sentiment analysis," *Fundations and Trends in information Retrieval*, vol. 2(1-2):1-135, 2008.

[8] S. Mohammad, "From once upon a time to happily ever after: Tracking emotions in novels and fairy tales," 2011.

[9] A. S. A. Naga, "A pagerank based detection technique for phishing web sites," *IEEE Symposium on Computers and Informatics*, 2012.

[10] N. Z. A. Pereira, "Syntactic similarity of web documents," 2003.

[11] A. Broder, "On the resemblance and containment of documents," *IEEE*, 1998.

[12] D. K. Keith Oatley and J. M. Jenkins, *Understanding emotions*, 2nd ed. Malden: Blackwell Publishing, 2006.

[13] F. S. P. Sargolzaei, "Pagerank problem, survey and future research directions," *International Mathematical Forum*, 2010.

[14] R. M. T. W. L. Page, S. Brin, "The pagerank citation ranking: Bring order to the web," *Stanford Digital Libraries Working Paper*, 1998.

[15] J. Kleinberg, "Authoritative sources in a hyperlinked enviroment," *ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[16] Keithhopper.com. (2012) Defining Community. [Online]. Available: https://marketplace.gephi.org/service/data-analysis/?gclid=CILShZz5n74CFS8V7AodYRQAsA

[17] C. Castillo, "Effective web crawling," *SIGIR Forum*, vol. 39, no. 1, pp. 55–56, Jun. 2005. [Online]. Available: http://doi.acm.org/10.1145/1067268.1067287

[18] S. Mohammad. (2014) Publications and Data emotion analysis. [Online]. Available: http://www.saifmohammad.com/WebPages/ResearchInterests.html

[19] C. Z. Y. Ou, "A new multimembership clustering method," *Journal of Industrial and Magnament Optimization*, 2007.

[20] Gephi.org. (2014) Gephi marketplace the official source for gephi plugins and services(beta). [Online]. Available: http://keithhopper.com/essay/definition-of-community

BIBLIOGRAPHY

[1] Art kleiner suicide notes database. URL http://www.well.com/~art/suicidenotes.html?w. (Cited on page 28.)

[2] The enron email corpus. URL http://www-2.cs.cmu.edu/~enron. (Cited on page 28.)

[3] *Graph theory with applications / J. A. Bondy and U. S. R. Murty.* Macmillan, 1976. ISBN 0333177916. (Cited on page 9.)

[4] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, pages 196–205. Springer, 2007. (Cited on page 29.)

[5] Bonka Boneva, Robert Kraut, and David Frohlich. Using e-mail for personal relationships the difference gender makes. *American behavioral scientist*, 45(3):530–549, 2001. (Cited on page 27.)

[6] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33 (1):309–320, 2000. (Cited on page 11.)

[7] Andrei Z Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE, 1997. (Cited on pages 11, 12, 13, and 20.)

[8] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011. (Cited on page 26.)

[9] Cyrus D Cantrell. *Modern mathematical methods for physicists and engineers*. Cambridge University Press, 2000. (Cited on page 42.)

[10] Xiaoyun Chen, Baojun Gao, and Ping Wen. An improved pagerank algorithm based on latent semantic model. In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, pages 1–4, Dec 2009. doi: 10.1109/ICIECS.2009.5364637. (Cited on pages 21 and 22.)

[11] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002. (Cited on page 17.)

[12] P. Gupta and K. Johari. Implementation of web crawler. In *Emerging Trends in Engineering and Technology (ICETET), 2009*

2nd International Conference on, pages 838–843, Dec 2009. doi: 10.1109/ICETET.2009.124. (Cited on pages 16 and 38.)

[13] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950. (Cited on page 14.)

[14] Hideaki Ishii, Roberto Tempo, and Er-Wei Bai. A web aggregation approach for distributed randomized pagerank algorithms. *Automatic Control, IEEE Transactions on*, 57(11):2703–2717, 2012. (Cited on pages 11 and 21.)

[15] Mario Jarmasz. Roget's thesaurus as a lexical resource for natural language processing. *arXiv preprint arXiv:1204.0140*, 2012. (Cited on page 26.)

[16] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989. doi: 10.1080/01621459.1989.10478785. (Cited on page 14.)

[17] Jaap Kamps, MJ Marx, Robert J Mokken, and Maarten De Rijke. Using wordnet to measure semantic orientations of adjectives. 2004. (Cited on page 29.)

[18] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics, 2006. (Cited on page 29.)

[19] Boo Vooi Keong and Patricia Anthony. Pagerank: A modified random surfer model. In *Information Technology in Asia (CITA 11), 2011 7th International Conference on*, pages 1–6. IEEE, 2011. (Cited on pages 11 and 21.)

[20] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999. (Cited on pages 11, 21, and 22.)

[21] Mason J. Knox. It Can Be Done: Poems of Inspiration, Cooperation. 1921. (Cited on page 7.)

[22] Donald E. Knuth. Computer Programming as an Art. *Communications of the ACM*, 17(12):667–673, December 1974. (Cited on page xiii.)

[23] Alfred L Kroeber. Statistics, indo-european, and taxonomy. *Language*, pages 1–21, 1960. (Cited on page 17.)

[24] V. I. Levenshtein. *Binary Codes Capable of Correcting Deletions, Insertions, and Reversals*, 10:707, 1966. URL www.scopus.com. Cited By (since 1996):1. (Cited on page 14.)

[25] Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman, and Slav Petrov. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174. Association for Computational Linguistics, 2012. (Cited on page 26.)

[26] Gang Liu, Bite Qiu, and Liu Wenyin. Automatic detection of phishing target from phishing webpage. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 4153–4156. IEEE, 2010. (Cited on pages 11, 15, 16, 25, and 26.)

[27] Wenyin Liu, Xiaotie Deng, Guanglin Huang, and Anthony Y Fu. An antiphishing strategy based on visual similarity assessment. *Internet Computing, IEEE*, 10(2):58–65, 2006. (Cited on pages 11, 13, 15, 16, 20, 25, 26, 61, and 62.)

[28] Frank E. Manuel. The Religion of Isaac Newton. 1974. (Cited on page 1.)

[29] Elhussein H Mohamed and M Nofal. Nrc publications archive (nparc) archives des publications du cnrc (nparc). (Cited on pages 13, 26, 29, 40, 41, and 64.)

[30] Saif Mohammad. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 105–114. Association for Computational Linguistics, 2011. (Cited on pages 13, 26, 29, 30, and 64.)

[31] Saif M Mohammad. From once upon a time to happily ever after: Tracking emotions in mail and books. *Decision Support Systems*, 53(4):730–741, 2012. (Cited on pages 13, 26, 29, 41, and 64.)

[32] Saif M Mohammad and Peter D Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34. Association for Computational Linguistics, 2010. (Cited on pages xvii, 13, 26, 27, 28, 29, 41, and 64.)

[33] Saif M Mohammad and Tony Wenda Yang. Tracking sentiment in mail: how genders differ on emotional axes. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (ACL-HLT 2011*, pages 70–79, 2011. (Cited on pages 13, 26, 27, 29, 41, and 64.)

[34] Sir Isaac Newton. Quote. 1760. (Cited on page 35.)

[35] Yongbin Ou and Cunquan Zhang. A new multimembership clustering method. *Journal of Industrial and Management Optimization*, 3(4):619, 2007. (Cited on pages 17, 18, 31, and 43.)

[36] L. Page, S. Brin, R. Motwani, and T. Winograd. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998. URL `citeseer.nj.nec.com/page98pagerank.html`. (Cited on pages 11, 20, 21, and 22.)

[37] AR Pereira and Nivio Ziviani. Syntactic similarity of web documents. In *Web Congress, 2003. Proceedings. First Latin American*, pages 194–200. IEEE, 2003. (Cited on pages 11, 13, 20, 25, and 26.)

[38] Robert Plutchik. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219, 1984. (Cited on pages 26, 28, 29, and 41.)

[39] Xingqin Qi, Kyle Christensen, Robert Duval, Edgar Fuller, Arian Spahiu, Qin Wu, and Cun-Quan Zhang. A hierarchical algorithm for clustering extremist web pages. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 458–463. IEEE, 2010. (Cited on pages 11 and 18.)

[40] Yan Qu, James Shanahan, and anyce Wiebe. *Exploring Attitude and Affect in Text, Theories and Applications: Papers from the 2004 AAAI Symposium: March 22-24, Stanford, California*. AAAI Press, 2004. (Cited on page 29.)

[41] Agustin Sancen-Plaza and Andres Mendez-Vazquez. Influence maximization for big data through entropy ranking and min-cut. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on*, pages 87–95. IEEE, 2013. (Cited on page 22.)

[42] Fernando Santos. Sentiment recognition for e-services. 2014. (Cited on pages 61 and 63.)

[43] P Sargolzaei and F Soleymani. Pagerank problem, survey and future research directions. In *International Mathematical Forum*, volume 5, pages 937–956, 2010. (Cited on pages 11, 20, 21, and 22.)

[44] Warren S Sarle. Algorithms for clustering data. *Technometrics*, 32 (2):227–229, 1990. (Cited on page 18.)

[45] Jitesh Shetty and Jafar Adibi. Discovering important nodes through graph entropy the case of enron email database. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 74–81, New York, NY, USA, 2005. ACM. ISBN 1-59593-215-1. doi: 10.1145/1134271.1134282. URL `http://doi.acm.org/10.1145/1134271.1134282`. (Cited on page 23.)

[46] A Naga Venkata Sunil and Anjali Sardana. A pagerank based detection technique for phishing web sites. In *Computers & Informatics (ISCI), 2012 IEEE Symposium on*, pages 58–63. IEEE, 2012. (Cited on pages 11, 25, and 26.)

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography *"The Elements of Typographic Style"*. `classicthesis` is available for both LaTeX and LyX:

`http://code.google.com/p/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured at:

`http://postcards.miede.de/`

Final Version as of October 24, 2014 (`classicthesis` version 4.2).

DECLARATION

Student's Declaration:

I Fernando Santos Sánchez, hereby declare that the work entitled "Sentiment & Emotion Recognition for Web Analysis." is my original work. I have not copied from any other students' work or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Guadalajara, October 2014

Fernando Santos

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Reconocimiento de sentimientos y emociones en el análisis de la web. Sentiment & Emotion Recognition for Web Analysis

del (la) C.

Fernando SANTOS SÁNCHEZ

el día 13 de Noviembre de 2014.

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González Pico
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Andrés Méndez Vázquez
Investigador CINVESTAV Guadalajara 2C
CINVESTAV Guadalajara