Stefan Eduard Raposo Alves

Nº 35098

Towards improving WEBSOM with Multi-Word Expressions

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : Nuno Cavalheiro Marques, Prof. Doutor,
Universidade Nova de Lisboa

Co-orientador : Joaquim Ferreira da Silva, Prof. Doutor,
Universidade Nova de Lisboa

Júri:

Presidente:	Henrique João
Arguente:	Carlos Ramisch
Vogal:	Nuno Cavalheiro Marques

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2013

Towards improving WEBSOM with Multi-Word Expressions

Abstract

Large quantities of free-text documents are usually rich in information and covers several topics. However, since their dimension is very large, searching and filtering data is an exhaustive task. A large text collection covers a set of topics where each topic is affiliated to a group of documents. This thesis presents a method for building a document map about the core contents covered in the collection.

WEBSOM is an approach that combines *document encoding* methods and Self-Organising Maps (SOM) to generate a document map. However, this methodology has a weakness in the document encoding method because it uses single words to characterise documents. Single words tend to be ambiguous and semantically vague, so some documents can be incorrectly related. This thesis proposes a new document encoding method to improve the WEBSOM approach by using multi word expressions (MWEs) to describe documents. Previous research and ongoing experiments encourage us to use MWEs to characterise documents because these are semantically more accurate than single words and more descriptive.

Keywords: Self-Organising Maps (SOM), Text Mining , WEBSOM, Relevant Expressions

Resumo

Uma enorme quantidade de documentos de texto é geralmente rica em informações e abrange diversos tópicos. No entanto, uma vez que a sua dimensão é muito volumosa, a pesquisa e filtragem de dados torna-se numa tarefa exaustiva. Um coleção grande de textos aborda um conjunto de temas onde cada tema é associado a um grupo de documentos. Esta tese apresenta um método para gerar um mapa de documentos sobre os principais conteúdos abordados numa coleção volumosa de textos.

A abordagem WEBSOM combina métodos de codificação de textos com Mapas Auto-Organizados (Self-Organising Maps) para gerar um mapa de documentos. Esta metodologia tem uma fraqueza na codificação de textos, pois utiliza apenas palavras singulares como atributos para caracterizar os documentos. As palavras singulares têm têndencia para ser ambíguas e semanticamente vagas e por isso alguns documentos são incorretamente relacionados. Esta dissertação propõe um método para codificar documentos através de atributos compostos por expressões relevantes. Através de estudos efetuados verificou-se que as expressões relevantes são bons atributos para caracterizar documentos, por serem semanticamente mais precisas e descritivas do que palavras singulares.

Palavras-chave: Self-Organising Maps (SOM), Text Mining , WEBSOM, Expressões Relevantes

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

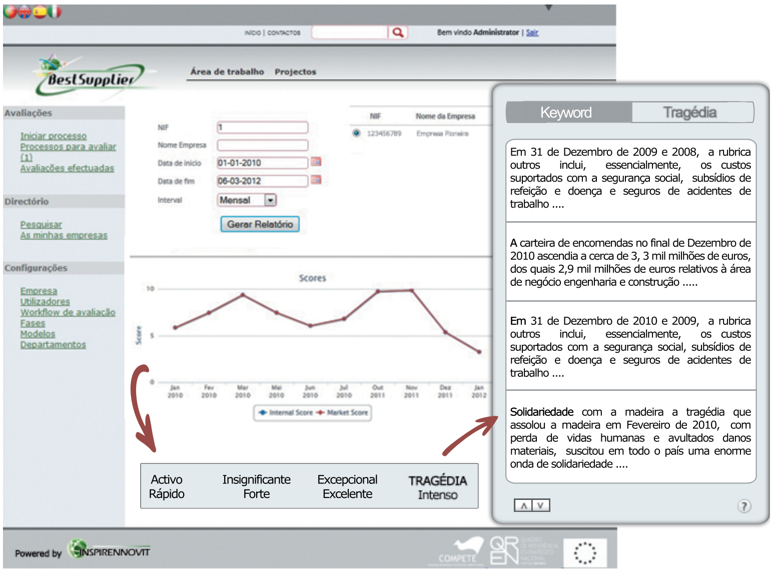
Large quantities of free-text documents can be seen as a large disorganised encyclopaedia. This text collection is rich in information but searching and filtering data is an exhaustive task. A large text collection covers a set of topics where each topic is affiliated to a group of documents. A document map about the core contents covered in the collection can be build by identifying the topics addressed in the documents. This is the main motivation for this thesis that aims to develop a document map that can relate documents not only by expressions in common but also with semantic associations. These associations occur when documents share the same topic but use different terms in their description, however, by identifying common patterns in documents, it is possible to relate them. Furthermore, a semantic map combined with a search engine responds faster in locating specific documents in answer to information requests, because the map has the documents organised and compressed by topics.

The emerging field of text mining applies methods to convey the information to the user in an intuitive manner. The methods applied vary according to the type of documents being analysed and the results that are intended. In this study, the different types of documents used are known as snippets, that are: parts of news, parts of reports and abstracts of articles. We intend to compose a *corpus* with these types of text and then build a context map about the core contents addressed in these documents.

The starting motivation for this dissertation was the need of building Corporate Semantic Maps in the context of the parallel ongoing project BestSupplier[1]. Best Supplier

[1] http://www.bestsupplier.eu, powered by the technological software development company InspirennovIT (http://www.madanparque.pt/pt/empresa/inspirennovit).

project needs to extract textual information related to companies. In the project framework, Self-organising maps were soon considered as a valuable tool for retrieving relations among documents and their summarisation. Figure 1.1 presents the initial information retrieval concept (as shown in [4]), that aimed to use snippets extracted form company reports (exemplified on the right of the figure) for mining related information and for building a Corporate Semantic Map. Latter, the Corporate Semantic Map could be used to relate concepts with documents. However, sharp semantic features were needed and the original WEBSOM soon showed its problems regarding a straight forward approach. This way, since the field studied by this project revealed to be too large to be analysed in this dissertation, a more academic perspective was chosen that yet, was still beneficial for the global project: trying to study how to improve WEBSOM quality by using better semantic aware features. The WEBSOM methodology has a weakness in the *document encoding* method because this one uses single words to characterise documents. Single words tend to be ambiguous and semantically vague, so some documents can be incorrectly related. This tends to be semantically less accurate than a well formed multi-word expressions.For example, the word "coordinates" is ambiguous because it has more than one meaning: *brings into proper place or order/a set of numbers used to calculate a position*. However, if "coordinates" it is taken within a relevant expression (RE), such as "map coordinates" or "coordinates the project", there is no ambiguity since it gains semantic sharpness. The fragility related to the poor accurate features used in the original WEBSOM builds up an important motivation of this thesis.



NIF: 1
Nome Empresa:
Data de início: 01-01-2010
Data de fim: 06-03-2012
Interval: Mensal

NIF	Nome da Empresa
123456789	Empresa Pioneira

Em 31 de Dezembro de 2009 e 2008, a rubrica outros inclui, essencialmente, os custos suportados com a segurança social, subsídios de refeição e doença e seguros de acidentes de trabalho ....

A carteira de encomendas no final de Dezembro de 2010 ascendia a cerca de 3, 3 mil milhões de euros, dos quais 2,9 mil milhões de euros relativos à área de negócio engenharia e construção .....

Em 31 de Dezembro de 2010 e 2009, a rubrica outros inclui, essencialmente, os custos suportados com a segurança social, subsídios de refeição e doença e seguros de acidentes de trabalho ....

Solidariedade com a madeira a tragédia que assolou a madeira em Fevereiro de 2010, com perda de vidas humanas e avultados danos materiais, suscitou em todo o país uma enorme onda de solidariedade ....

Figure 1.1: Example of the system interaction with WEBSOM.

1.2 Objectives

The main focus of this thesis is to develop a document map about topics addressed in documents, using self-organising maps. The WEBSOM addresses this field applying text-mining methods, however, this approach uses single words to characterise documents. Our goal is to extend the WEBSOM module in a manner that documents are characterised by relevant expressions (REs), because these are semantically more accurate than single words. The following requirements are the goals that are intended to accomplish:

Goal 1. Extract the snippets content by their REs:

Snippets are composed by textual data with some dimension. This text size can be compressed into a set of REs that describe the main content of it. So, we intend to learn how to extract all possible meaningful expressions that describe the documents core content.

Goal 2. Measure and weight REs:

As it was said before, the content of a snippet is compressed in a set of REs. Although the description of the snippet is obtained according to that set, there are expressions that are more informative than others. In this goal a measure to weight the relevance of each RE in the *corpus* is developed. By measuring the terms extracted in **goal 1**, it is possible to determine which are those that are more influential in the collection. For instance, an expression may be a relevant and well composed multi-word, but if it occurs very often in the majority of documents, it is not very influential. Besides, since the extractor used in **goal 1** also extracts some incorrect REs, not all of them are well formed multi-word expressions and it is necessary to penalise these ones.

Goal 3. Reducing the document features dimension:

The documents in text-mining are usually represented as a vector of the size of the vocabulary where each position corresponds to some function of the frequency of a particular keyword in the document. For this study, keywords are REs extracted in **goal 1**, which results in a vast dimensionality of the document vectors because the number of REs in a *corpus* easily rises to tens of thousands. Typically, a huge number of features is computationally heavy for clustering algorithms, besides, the results may lose precision. So, in order to reduce the number of features, this study proposes to represent documents in a *document-by-document* vector where each document is characterised by its similarities with all the other documents. Similarities are calculated considering all REs in the vocabulary.

Goal 4. Develop a semantic map using WEBSOM:

The original WEBSOM approach uses self-organised maps(SOM) to automatically organise a textual collection onto a two-dimensional projection designated as context map. In order to extend the WEBSOM module with REs it is necessary to combine the structure produced in **goal 3** with the SOM. So, this dissertation intends to study the WEBSOM module to build a document map from the structure produced in the previous goal.

Goal 5. Combine the semantic map with REs:
The semantic map reflects an organised collection of documents, where documents with similar attributes are close in the map and dissimilar documents are in different areas. Although this structure is useful, it is necessary to understand why the documents are together in a intuitive manner. Since the map has similar documents placed in the same area, it is possible to determine the main REs responsible for that document proximity in the map, and then represent these locations with them. So, as meaningful REs give a strong perception of the main core content of documents, a method is proposed to visualise the impact of REs in the semantic map.

1.3 Structure of the thesis

Chapter 2 is the result of a literature study that forms the basis of this thesis. A brief overview about clustering algorithms and their application to document clustering is done. Then, the WEBSOM method and keyword extractor is presented. Chapter 3 proposes an approach, designated as RSOM, for the document clustering method in the context of document Information Retrieval techniques. This chapter also outlines the architecture chosen for this approach. Chapter 4 evaluates and exemplifies the application of the proposed methodology . Finally, several conclusions are drawn regarding the RSOM.

2

Related Work

In this Chapter a study about different types of text-mining algorithms to conduct this dissertation is presented. First, Section 2.1 performs an analysis about existing clustering methodologies. Then, Section 2.2 presents an approach to represent specific data such as free-text for document clustering algorithms. Section 2.3 describes the WEBSOM module and identifies the components which have to be modified to integrate REs. Moreover, why the REs can be a solution to solve the semantic fragility of individual words is also discussed. The following Section 2.4 presents a detailed description about a REs extractor used in this study.

2.1 Clustering

2.1.1 Hierarchical clustering

Hierarchical clustering [5] is a data-mining algorithm which aims to cluster objects in such a manner that in a cluster objects are more similar to each other than instances in different clusters. As the name suggests, clusters are formed hierarchically according to a "bottom up" or "top down" approach. In other words, there are two ways to form hierarchical cluster:

- **Agglomerative clustering** is a "bottom up" approach which begins with a cluster for each training object, merging pairs of similar clusters as it moves into higher levels of the hierarchy, until there is just one cluster.

- **Divisive clustering** is a "top down" approach which begins with one large cluster and divides into smaller clusters as it moves lower in the hierarchy, until there is one cluster for each training object.

Both agglomerative and divisive clustering determine the clustering in a *greedy algorithm*[1]. However, the complexity for divisive is worse than agglomerative clustering, which is $\mathcal{O}(2^n)$ and $\mathcal{O}(n^3)$ respectively, where n is the number of training objects. Furthermore, in some cases of agglomerative methods a better complexity of $\mathcal{O}(n^2)$ has been accomplished, namely: *single-link*[6] and *complete-link*[7] clustering.

Measure Name	**Measure Formula**
Euclidian distance	$d(x,y) = \sqrt[2]{\sum_i (x_i - y_i)^2}$
Manhattan distance	$d(x,y) = \sum_i \lvert x_i - y_i \rvert$
Maximum distance	$d(x,y) = \max_i \lvert x_i - y_i \rvert$
Cosine similarity	$d(x,y) = \frac{x \cdot y}{\lVert x \rVert \lVert y \rVert}$

Table 2.1: Some common measures used in hierarchical clustering.

In *single-link clustering*, the distance is defined as the smallest distance between all possible pair of objects of the two clusters:

$$d(\mathcal{C}_x, \mathcal{C}_y) = \min_{x \in \mathcal{C}_x, y \in \mathcal{C}_y} d(x,y)$$

where $d(x,y)$ is a generic distance measure between object x and y. Some common measures used in hierarchical clustering are displayed in table 2.1. In *complete-link clustering*, the distance between two clusters is taken as the largest distance between all possible clusters:

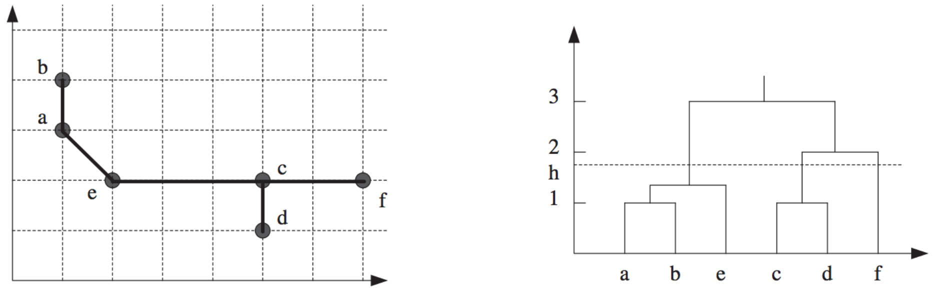
$$d(\mathcal{C}_x, \mathcal{C}_y) = \max_{x \in \mathcal{C}_x, y \in \mathcal{C}_y} d(x,y).$$

The previous methods are most frequently used to choose the two closest groups to merger. There are other methods such as the *average-link* method that uses the average of distances between all pairs and the centroid[2] distance between two clusters.

The result obtained using a hierarchical clustering algorithm is generally expressed as a *dendrogram*. An example is given in figure 2.1 with the *single-link* method, where the lower row in figure 2.1(b) represents all instances, and as we go into higher levels, clusters are merged according to their similarities. The h value represents the hierarchy level that determines the number of clusters, which is the minimal number of clusters having all objects with distance lower than h. Although *complete-link* forms clusters differently

[1] A greedy algorithm is an algorithm that solves the problems according to a heuristic to make the local optimal choice at each stage.

[2] The cluster centroid means the centre of the cluster in a d-dimension.



(a) A two-dimensional data set.

(b) The dendrogram result.

Figure 2.1: An agglomerative clustering using the *single-link* method.

from *single-link*, the number of clusters is determined in the same manner. In general, this requirement occurs in both agglomerative and divisive clustering. The value of h is specified by the user, which makes this algorithm inadequate for aim of this dissertation because the number of clusters should be formed according to object attributes in a *natural way*.

2.1.2 K-means and k-medoids

K-means [8] is a data-mining algorithm which aims to distribute data objects in a defined number of k cluster where each object belongs to the cluster with nearest centroid. This clustering results in a partition of the data space into *Varonoi diagrams*[9]. In other words for a dataset $X = (x_1, x_2, ..., x_n)$ in which x_i is a d-dimensional real vector, k-means aims to partition the n objects into k clusters $\mathcal{C} = (c_1, c_2, ..., c_k)$ so as to minimise the within cluster sum of squares:

$$\arg\min_{\mathcal{C}} \sum_{i=1}^{k} \sum_{x_j \in X} \|x_j - \mu_i\|^2$$

where μ_i is the centroid of cluster i.

An inconvenience of k-means lies on the fact that the objects distance is measured by the Euclidean distance because it is assumed that the clusters are spheroids with the same size, which does not occur in most real cases. In fact, in most clustering configurations, each cluster presents its own volume, shape and orientation. An example of this behaviour can be visualised in figure 2.2 that is a k-means clustering result for the *Iris flower data set*[3][10] and the real species distribution using the *ELKI*[4] framework, in which the cluster centroid is marked using a larger symbol.

K-medoids clustering [11] is a partitioning algorithm similar to k-means algorithm. In contrast, instead of building centroids, this one chooses objects from the data set as cluster centres (medoids or exemplars). This method is more robust to noise and outliers in comparison to k-means because it minimises the sum of pairwise dissimilarities

[3] The Iris flower dataset is a data collection about variation of Iris flowers of three related species.

[4] Environment for DeveLoping KDD-Applications Supported by Index-Structures Framework, http://elki.dbs.ifi.lmu.de/

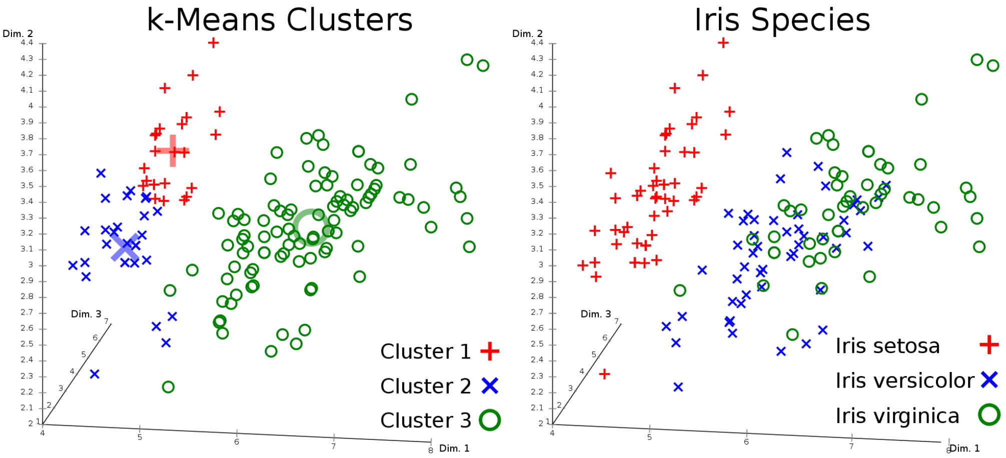


Figure 2.2: K-means clustering result for the *Iris flower data set* and the real species distribution using *ELKI*.

instead of the sum of squared Euclidean distances.

The number of clusters has to be specified in both k-means and k-medoids. This requirement makes these algorithms inadequate for a fully unsupervised context where we want to the methodology to tell us how many clusters there are, among other information such as the distribution of the objects in these clusters. Both clustering methods are inadequate to be included as a tool in this thesis approach because the specification of the number of clusters should not be a requirement of the system. Instead, the output of the system should suggest how documents are grouped.

2.1.3 The Model-Based Cluster Analysis

Taking into account that K-means and k-medoids requires the explicit specification of the number of clusters, the Model-Based Clustering Analysis is an alternative approach to solve this limitation.

Thus, considering the problem of determining the structure of clustered data, without prior knowledge of the number of clusters or any other information about their composition, Fraley and Raftery [12] developed the Model-Based Clustering Analysis (MBCA). By this approach, data is represented by a mixture model where each element corresponds to a different cluster. Models with varying geometric properties are obtained through different Gaussian parameterizations and cross-cluster constraints. This clustering methodology is based on multivariate normal (Gaussian) mixtures. So the density function associated to cluster c has the form

$$f_c(\vec{x}_i|\vec{\mu}_c,\vec{\Sigma}_c) = \frac{exp\{-\frac{1}{2}(\vec{x}_i-\vec{\mu}_c)^T\vec{\Sigma}_c^{-1}(\vec{x}_i-\vec{\mu}_c)\}}{(2\pi)^{\frac{r}{2}}\left|\vec{\Sigma}_c\right|^{\frac{1}{2}}} \quad . \tag{2.1}$$

Clusters are ellipsoidal, centred at the means $\vec{\mu}_c$; element $\vec{x}_i$ belongs to cluster c. The

$\vec{\Sigma}_c$	Distribution	Volume	Shape	Orientation
$\lambda\vec{I}$	Spherical	Equal	Equal	
$\lambda_c\vec{I}$	Spherical	Variable	Equal	
$\lambda\vec{D}\vec{A}\vec{D}^T$	Ellipsoidal	Equal	Equal	Equal
$\lambda_c\vec{D}_c\vec{A}_c\vec{D}_c^T$	Ellipsoidal	Variable	Variable	Variable
$\lambda\vec{D}_c\vec{A}\vec{D}_c^T$	Ellipsoidal	Equal	Equal	Variable
$\lambda\vec{D}_c\vec{A}\vec{D}_c^T$	Ellipsoidal	Variable	Equal	Variable

Table 2.2: Parameterizations of the covariance matrix $\vec{\Sigma}_c$ in the Gaussian model and their geometric interpretation

covariance matrix $\vec{\Sigma}_c$ determines other geometric characteristics. This methodology is based on the parameterisation of the covariance matrix in terms of eigenvalue decomposition in the form $\vec{\Sigma}_c = \lambda_c\vec{D}_c\vec{A}_c\vec{D}_c^T$, where $\vec{D}_c$ is the matrix of eigenvectors, determining the orientation of the principal components of $\vec{\Sigma}_c$. $\vec{A}_c$ is the diagonal matrix whose elements are proportional to the eigenvalues of $\vec{\Sigma}_c$, determining the shape of the ellipsoid. The volume of the ellipsoid is specified by the scalar λ_c. Characteristics (orientation, shape and volume) of distributions are estimated from the input data, and can be allowed to vary between clusters, or constrained to be the same for all clusters. The input data is an input matrix where objects (documents) are characterised by features.

MBCA subsumes the approach with $\vec{\Sigma}_c = \lambda\vec{I}$, long known as *k-means*, where the sum of squares criterion is used, based on the assumption that all clusters are spherical and have the same volume (see Table 2.2). However, in the case of k-means, the number of clusters has to be specified in advance, and considering many applications, real clusters are far from spherical in shape.

During the cluster analysis, *MBCA* shows the Bayesian Information Criterion (BIC), a measure of evidence of clustering, for each pair *model-number of clusters*. These pairs are compared using BIC: the larger the BIC, the stronger the evidence of clustering (see [12]). The problem of determining the number of clusters is solved by choosing the *best model*. Table 2.2 shows the different models used during the calculation of the *best model*.

Although the MBCA solves the number of clusters specification requirement of k-means and k-medoids, due to the complexity of this approach, it is very sensitive to the number of attributes used to characterise the input objects, ex: if this number is greater than 20 features for some hundreds of objects the system becomes computationally heavy. Since one of the focuses of the present dissertation is to compose a topic context map, the MBCA clustering is inadequate, because this method doesn't provide any topological relation between clusters.

2.1.4 Fuzzy Clustering

Contrary to other clustering approaches where objects are separated in different groups, such that each object belongs to just one cluster, in fuzzy clustering (also referred to as soft clustering), each object may belong to more than one cluster. Moreover, each object is associated to a set of membership levels concerning the clusters. These levels rule the strength of association between each object and each particular cluster. Thus, fuzzy clustering consists in assigning these membership levels between objects and clusters. After clusters are created, new objects may be assigned to more than one cluster.

Fuzzy C-Means Clustering (FCM) (Bezdec 1981) is one of the widely used fuzzy clustering algorithms. FCM aims to partition the set of n data objects $X = \vec{x_1}, \ldots, \vec{x_n}$ into a set of fuzzy clusters under some criterion. Thus, for a set of objects, X, FCM finds a set of c fuzzy clusters $C = c_1, \ldots, c_c$ and it also returns a fuzzy partition matrix, $U = [u_{i,j}]$, where each cell reflects the *belonging degree* of each object j to each cluster i, that is $u_{i,j}$, where $u_{i,j} \in [0, 1]$, $i = 1, \ldots, c$, $j = 1 \ldots, n$.

The FCM algorithm minimises a function such as the following generic objective function:

$$J(X, U, B) = \sum_{i=i}^{c} \sum_{j=1}^{n} u_{ij}^{m} d^2(\vec{\beta_i}, \vec{x_j}) \tag{2.2}$$

subject to the following restrictions:

$$(\sum_{j=1}^{n} u_{ij}) > 0 \; \forall \; i \in 1, \ldots, c \text{ and} \tag{2.3}$$

$$(\sum_{i=1}^{c} u_{ij}) = 1 \; \forall \; j \in 1, \ldots, n \tag{2.4}$$

where $\vec{\beta_i}$ is the prototype of cluster c_i (in the case of FCM, it means no extra information but the cluster centroid), and $d(\vec{\beta_i}, \vec{x_j})$ is the distance between object $\vec{x_j}$ and the prototype of cluster c_i. B is the set of all c cluster prototypes $\vec{\beta_1}, \ldots, \vec{\beta_c}$. Parameter m is called the fuzzifier and determines the *fuzziness* of the classification. Higher values for m correspond to softer boundaries between the clusters; with lower values harder boundaries will be obtained. Usually $m = 2$ is used.

Constraint 2.3 ensures that no cluster is empty; restriction 2.4 guarantees that the sum of the membership degrees of each object equals 1.

For the membership degrees, the following must be calculated:

$$u_{ij} = \begin{cases} \frac{1}{\sum_{k=1}^{c} (\frac{d^2(\vec{x_j}, \vec{\beta_i})}{d^2(\vec{x_j}, \vec{\beta_k})})^{\frac{1}{m-1}}}, & : \quad I_j = \emptyset \\ 0, & : \quad I_j \neq \emptyset \text{ and } i \notin I_j \\ x, \; x \in [0, 1] \text{ such that } \sum_{i \in I_j} u_{ij} = 1, & : \quad I_j \neq \emptyset \text{ and } i \in I_j \; . \end{cases} \tag{2.5}$$

Equation 2.5 shows that the membership degree of an object to a cluster depends not only on the distance between the object and that cluster but also from the object to all other clusters.

Although fuzzy clustering may reflect some real cases because sometimes objects belong to more than one group, this clustering approach does not create any map of contexts. This dissertation aims to represent documents in a map where distances are ruled by semantic categories; this environment is not available in fuzzy clustering algorithms.

2.1.5 Self-Organising Maps

The self-organising maps (SOM) is an algorithm proposed by Teuvo Kohonen[13, 14], that performs an unsupervised training of an artificial neural network. In Kohonen's model, the neurones learn how to represent the input data space onto a two-dimensional map, in which similar instances of the input data are close to each other. In other words, it can also be said that SOM reshapes the input dimension into a plain geometric shape so that important topological relations are preserved.

The training algorithm is based on *competitive learning*, in which neurones from the network compete to react for certain patterns of the data set. For each neurone i a parametric *model / weight vector* $m_i = [\mu_{i,1}, \mu_{i,2}, \ldots, \mu_{i,n}]$ is assigned, where n stands for the number of attributes of the input dimension. In the original SOM algorithm the model vector components are initialised with random values as a starting point, and as the training process follows the model vectors shape into a two-dimensional ordered map. At the beginning of the learning process an input vector x_j is fed to the network, then the similarity between vector x_j and all model vectors is measured. The neurone with the most similar weight vector to the input is denoted as the *best-matching unit* (BMU), the similarity value is usually calculated by the Euclidean distance:

$$\|x - m_c\| = \min_i\{\|x - m_i\|\} \tag{2.6}$$

Once the BMU m_c is found, this one is updated so that its values resemble more the attributes of vector x_j. Neurones which are close to neurone m_c also have their weights affected by the input vector. The surrounding update results in a local *relaxation effect* which in ongoing training leads to the global neurones organisation. The update method for the model vector m_c is

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]\ , \tag{2.7}$$

where t means the iteration number and $x(t)$ the selected input vector at that iteration. The $\alpha(t)$ represents the *learning-rate factor*, taking real values between $[0, 1]$. The function $h_{ci}(t)$ stands for the *neighbourhood function*, which is responsible for the relaxation effect. This impact occurs in one area of the map, so only neurones that are in the range of the BMU are affected. This neighbourhood radius is usually determined by two variants: the map distance between the coordinates of the BMU, that is r_c, and the coordinates of neurone i, that is r_i ; and a parameter $\sigma(t)$ that gradually increases during each iteration. In the original SOM, the neighbourhood function is estimated by the Gaussian function,

$$h_{ci}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right). \tag{2.8}$$

An example of a local update is shown in figure 2.3. The black dot represents the BMU m_c of an input vector x_j, and the relaxation effect is demonstrated by the colour intensity of the surrounding neurones. As is shown in the figure, the neurones closer to the BMU a have higher learning rate and as it goes further from centre the learning rate decreases.

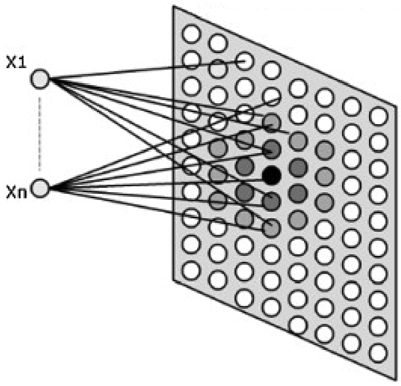


Figure 2.3: Example of an input vector update when fed to the network.

The idea is to train the neurones in two phases: in the first phase there is a large learning factor and neighbourhood radius, which contributes for a larger-scale approximation to the data; then as iterations follow, both learning factor and neighbourhood radius gradually decrease into a fine tuned phase, where the learning factor and neighbourhood radius is small. As the learning process finishes, the model vectors are trained to reflect a two-dimension organised map.

The *batch training* is a faster alternative to train Kohonen's neural networks proposed by Yizong Cheng [15]. In this training, the data set is partitioned into n observations sets $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, then at each learning step, instead of training with single input vector x_j the network is fed by the average weight $\bar{x}_i$ of an observation set s_i. Additional details are presented at [15], in which the proof about *topological order* is presented. Furthermore, in the *SOM Toolbox* [14, 16] which is an implemented SOM package for Matlab, the complexity gain when using batch training is shown.

There are several visual methods to analyse the map topology and neurones weight attributes. In order to familiarise with the SOM algorithm and respective visual tools, a case study about the *Iris flower data set* [10] is shown bellow, using the SOM Toolbox package. The *U-Matrix* [17] proposes a distance based map, meant to visualise the distance between neurones of the network. An example of a U-Matrix is presented at figure

2.4. The neurones are represented as hexagons, and for each pair of neurones another hexagon is placed, which reflects their distance . The colour bar at the right of the map represents the distance scale values.

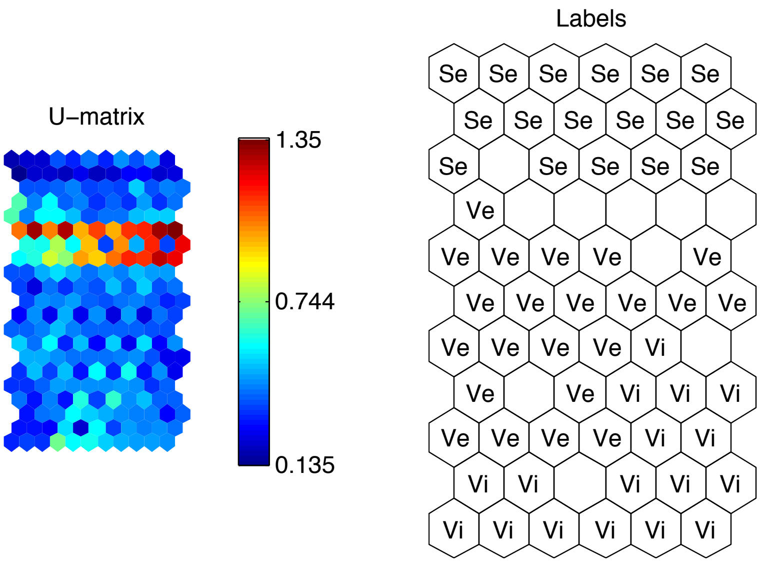


Figure 2.4: U-matrix and respective labelling using the iris data set.

The figure demonstrates a characteristic of the U-Matrix that is the visual identification of clusters. In this case, it is clear the top region has a red line (which represents high distance values) of hexagons that sets a border between the specie *setosa* and *versicolor*. The same cannot be said between *versicolor* and *virginica* that seem to collide in some cases. In summary the U-Matrix gives a perception about the neurones distance between them, that for this study may become useful to verify the distant result of an organised document map. The component plane is a method to visualise the neurones weight values of a specific attribute. For instance, in figure 2.5 illustrates the weight values map for each Iris attribute. This visual is often matched with the U-Matrix, in order to identify

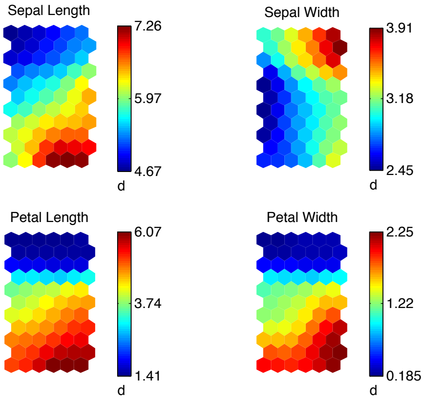


Figure 2.5: Component Planes for the Iris data set.

the attribute values in different areas of the map.

There are other visual methods to extract information about the neural network. For instance in [18, 19, 20] illustrates several types of projections, where each of them displays distinct information about the same network. Some of these tools can be used during the dissertation for an analysis over the document map.

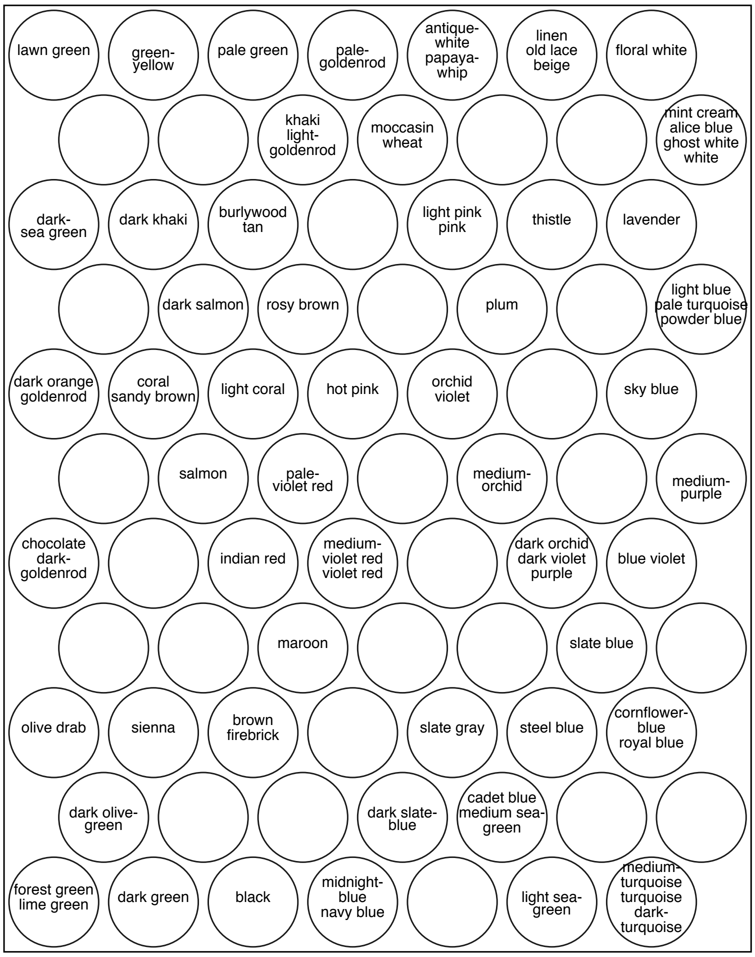


Figure 2.6: A map of colours based on their RGB values.

For the present thesis, we intended to achieve a result similar to the map colours presented in [21], where a neural network is trained to project an organised map about different types of colours. In figure 2.6 we illustrate the projection result, where similar colours are either close or located in the same area. There are many research articles [22] about SOM that use it as a tool to solve scientific and real-word problems. In section 2.3, the combination of textual data with of the SOM algorithm, known as WEBSOM [23], is analysed.

Self-organised maps handles well a high-dimensional space, which occurs in the documents characterisation. This algorithm matches the document map goal of this study, because the documents are projected into a context map in which they are organised and grouped according to their topics. However, this algorithm requires us to prepare the documents into a supported input structure. In the next section, document data representation and document clustering for textual-data is analysed.

2.2 Document Clustering

Document clustering finds applicability for a number of tasks: in document organisation and browsing, for efficient searching; in corpus summarisation, providing summary insights into the overall content of each cluster of the underlying corpus; and in document classification, since after clusters are built during a training phase, new documents can be classified according to the clusters that were learnt.

Many classes of algorithms such as the k-means algorithm, or hierarchical algorithms are general-purpose methods, which can be extended to any kind of data, including text data. A text document can be represented either in the form of binary data, when using the presence or absence of a word in the document in order to create a binary vector. In such cases, it is possible to directly use a variety of categorical data clustering algorithms [24, 25, 26] on the binary representations. Although, a more enhanced representation may include refined weighting methods based on the frequencies/probabilities of the individual words in the document as well as frequencies of words in an entire collection (e.g., TF-IDF weighting [27]). Quantitative data clustering algorithms [28, 29, 30] can be used in conjunction with these weightings in order to determine the most relevant terms in the data to form clusters.

Naive techniques do not typically work well for clustering text data. This is due to the fact that text representation data uses a very large number of distinguishing characteristics. In other words, the lexicon associated to the documents may be of the order of 10^5, though a single document may contain only a few hundred words. This problem may be more serious when the documents to be clustered are very short such as tweets, for example.

While the lexicon of a given corpus of documents may be large, the words are typically correlated with one another. This means that the number of concepts (or principal components) in the data is much smaller than the feature space. This demands for algorithms that can account for word correlations, such as [31] for example. Nevertheless, if only single words are used, part of the semantic sharpness of the document content is lost, which may lead to clustering errors. As an example, since the word "coordinates" has more than one meaning (e.g. *he coordinates the project* vs *map coordinates*) it may leads to serious deviations in the calculation of the correlation values and therefore some incorrect cluster formation. However, the semantic sharpness associated to text attributes can be greatly improved if multi-word expressions/relevant expressions were used to characterise documents [31]: "map coordinates" instead of "coordinates", or "bus stop" and "bus address", instead of "bus".

The sparse and high dimensional representation of the different documents has also been studied in the information retrieval literature where many techniques have been proposed to minimise document representation for improving the accuracy of matching a document with a query [27].

2.2.1 Vector space model

In general, a common representation used for text processing is the vector-space model (VSM). In the basic version of this model [32], a document is stored as binary vector where each attribute of the vector corresponds to words of the vocabulary, and the value of an attribute is 1 if the respective word exists in the document; otherwise it is 0. Although the binary vector encoding may be useful in certain frameworks, in other cases it is possible to replace them by real values through a function of the word occurrence frequency that results in a statistical value about the word influence in the document. For that, the *vector-space based* Tf-Idf can be used.

2.2.1.1 Tf-Idf

Tf-Idf (Term frequency−Inverse document frequency) is a statistical metric often used in Information Retrieval (IR) and Text Mining to evaluate how important a term W (word or multi-word) is to a document d_j in a corpus $\mathcal{D}$. It has the following expression:

$$Tf\text{–}Idf(W, d_j) = \frac{f(W, d_j)}{size(d_j)} \,.\, \log \frac{\|\mathcal{D}\|}{\|\{d : W \in d\}\|} \tag{2.9}$$

$\|\mathcal{D}\|$ stands for the number of documents of corpus $\mathcal{D}$; $\|\{d : W \in d\}\|$ means the number of documents containing term W, and $size(d_j)$ is the number of words of d_j. Some authors prefer to use the probability ($f(W, d_j)/size(d_j)$) of term W in document d_j instead of the more commonly used absolute frequency ($f(W, d_j)$), as bias towards longer documents can be prevented.

2.2.2 Feature Selection and Transformation Methods for Text Clustering

2.2.2.1 Stop Words Removal

The quality of any data mining method such as classification and clustering is highly dependent on the noisiness of the features used in the clustering process. Most authors discard commonly used words such as "the", "by", "of" to reduce the number of features, thinking they are useless to improve clustering quality. However, these *stop words* are useful when taken as part of meaningful multi-words, e.g. "President of the Republic".

2.2.2.2 Entropy-based Ranking

Setting different weights to words is also used for feature selection in text. The entropy-based ranking was proposed in [33]. The quality of the term is measured by the entropy reduction when it is removed. The entropy $E(t)$ of the term t in a collection of n documents is defined by:

$$E(t) = -\sum_{i=1}^{n}\sum j = 1^n(S_{ij} \,.\, \log(S_{ij}) + (1 - S_{ij}) \,.\, \log(1 - S_{ij}) \tag{2.10}$$

where $S_{ij} \in [0, 1]$ is the similarity between the ith and jth document in the collection of n documents is defined as follows:

$$S_{ij} = 2^{-\frac{dist(i,j)}{\overline{dist}}} \quad . \tag{2.11}$$

$dist(i, j)$ stands for the distance between the terms i and j after the term t is removed; $\overline{dist}$ is the average distance between the documents after term t is removed.

2.2.2.3 Dimensionality Reduction Methods

Feature transformation is a different method in which the new features are defined as a functional representation of the original features. The most common is the dimensionality reduction method [34] in which the documents are transformed to a new feature space of smaller dimensionality where features are typically a linear combination of the original data features. Methods such as Latent Semantic Indexing (LSI) [35] use this common principle. The overall effect is to remove many dimensions in the data which are noisy or partially redundant for similarity based applications such as clustering. LSI is closely related to the problem of Principal Component Analysis (PCA). For a d-dimensional data set, PCA builds a symmetric $d \times d$ covariance matrix C of the data, such that each (i, j)th cell of C is the covariance between dimensions i and j. This matrix is positive semi-definite and can be decomposed as follows:

$$C = P \,.\, D \,.\, P^T \tag{2.12}$$

P is a matrix whose columns contain orthonormal eigenvectors of C and D is a diagonal matrix containing the corresponding eigenvalues. The eigenvectors represent a new orthonormal basis system along which the data can be represented. Each eigenvalue corresponds to the variance of the data along each new dimension. Most of the global variance is preserved in the largest eigenvalues. Therefore, in order to reduce the dimensionality of the data, a common approach is to represent the data in this new basis system, which is further truncated by ignoring those eigenvectors for which the corresponding eigenvalues are small. LSI is quite similar to PCA except that uses it approximation of the covariance matrix C which is quite appropriate for the sparse and high-dimensional nature of text data. In other words, let A be a $n \times d$ term-document matrix in which the (i, j)th entry is the normalised frequency for term j in document i. It can be shown that $A^T{\cdot}A$ is a $d \times d$ matrix and it would be the same as a scaled version (by factor n) of the covariance matrix, if the data is mean-centred. As in the case of numerical data, LSI uses eigenvectors of $A^T{\cdot}A$ with the largest variance in order to represent the text. In typical collections, only about 300 to 400 eigenvectors are required for the representation. If original attributes used in LSI are based only on single words frequencies, the finer granularity of the words polysemy may be lost.

The random mapping [36] is a faster reduction method, compared to LSI and PCA, to reduce a high-dimensional onto a much lower-dimensional space. However, this approach usually has a greater loss of information than LSI and PCA. The random projection is formed by multiplying the original matrix M by a random matrix R,

$$N = M\,.\,R\,, \tag{2.13}$$

where N is the reduced projection and its dimensionality is much smaller than the original input dimensionality. The random matrix sets for each column a fixed number of randomly distributed ones and the rest of the elements are zeros. The size of random features varies depending on the original matrix, in which the lower-dimension has to be sufficiently orthogonal to provide an approximation of the original features. Although the random mapping method may seem simple it has been successful to reduce the documents high-dimension into a lower-dimension space [37, 38], to later be organised onto a document map. Further details concerning random mapping properties are presented in [36].

2.2.3 Distance-based Clustering Algorithms

Distance-based clustering algorithms are designed by using a similarity function to measure the similarity between text objects. One of the most popular measures is the $cosinesimilar$ function:

$$cosine(U, V) = \frac{\sum_{i=1}^{k} f(u_i\,.\,f(v_i}{\sqrt{\sum_{i=1}^{k} f(u_i)^2}\,.\,\sqrt{\sum_{i=1}^{k} f(v_i)^2}} \tag{2.14}$$

where $U = (f(u_1) \ldots f(u_k))$ and $V = (f(v_1) \ldots f(v_k))$. Other similarity functions include Pearson correlation and other weighting heuristics and similarity functions. TF-IDF is used in [39]; BM25 term weighting is used in [40].

Hierarchical Clustering-based Algorithms are used in the context in text data in [41, 42, 43, 44, 45, 46, 47, 48]. Limitations associated to Hierarchical Clustering algorithms are explained in section 2.1.1.

Distance-based Partitioning Algorithms are widely used: k-medoid and k-means clustering are used for text context in [49] and [50], for example. The work in [51] uses the frequency of phrases as text attributes.

2.3 WEBSOM

WEBSOM [52, 53] aims to automatically organise an arbitrary free-text collection to convey browsing and exploration of the documents. The method consists of two hierarchical connected SOMs: in the first SOM, the word histograms[5] used to characterise documents are encoded according to their contexts, so that a *word category map* [54] is developed to compose word categories (further details in section 2.3.1); for the second level, the documents are encoded according to the word categories developed in the previous level, and then clustered using the SOM algorithm. Figure 2.7 presents an overview of the original WEBSOM architecture.

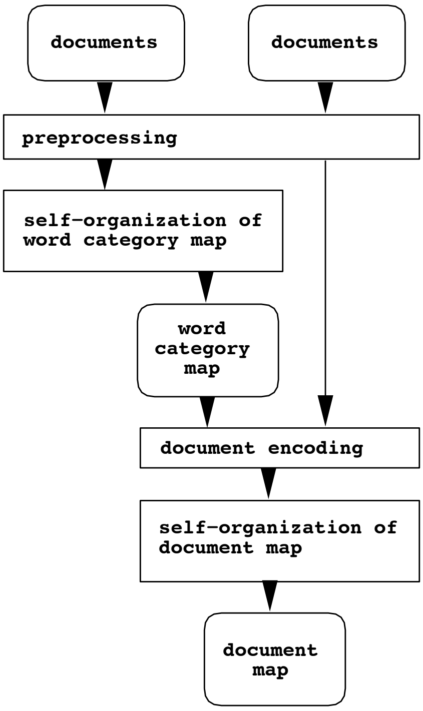


Figure 2.7: The original WEBSOM architecture.

2.3.1 Word Category Map

Documents tend to use different terms to describe the same topic. This occurs because each author has his own writing style, so when they compose a document, the topics are described in their manner. This results in a large vocabulary dimension and as documents are represented by the VSM (section 2.2.1), expressions with similar meaning are indexed as distinct from one to another. The WEBSOM encodes words according to their average context, in order to develop a SOM about word categories [21, 55]. The average context is

[5] The word histogram is the list of words used to characterise a document.

represented by the vector ${x_i}^{(d)}$, which denotes a statistical description of the words that occur at neighbourhood d from the word i. For a set of displacements $\{d_1, \ldots, d_n\}$, the average context of word i is

$$x_i = \begin{bmatrix} {x_i}^{d_1} \\ \vdots \\ {x_i}^{d_n} \end{bmatrix} . \tag{2.15}$$

Usually the contextual information is computed for two displacements, the preceding word ${x_i}^{-1}$ and succeeding word ${x_i}^{+1}$. There are different possibilities to encode the word context ${x_i}^{d}$. For instance in [56], the context of word i at the distance d is encoded as the vector

$${x_i}^{(d)} = \frac{1}{|{I_i}^{(d)}|} \sum_{k \in {I_i}^{(d)}} e_k , \tag{2.16}$$

where e_k stands for the unit vector of the word k. ${I_i}^{(d)}$ is the set of words at distance d from word i and $|{I_i}^{(d)}|$ means the size of this set. This encoding has an inconvenience due to the high dimensionality of the vocabulary. In order to reduce the dimension, the unit vector e_k is replaced by a r-dimensional random vector r_k (section 2.2.2.3), which results in

$${x_i}^{(d)} = \frac{1}{|{I_i}^{(d)}|} \sum_{k \in {I_i}^{(d)}} r_k , \tag{2.17}$$

where the r dimension has to be sufficiently orthogonal to provide an approximation of the original basis. For instance, for the case study [2] it reduced from 39 000 to a 1000 dimensions. Another study at [56] reduced the dimension from 13 432 to 315. So as we can see the dimension varies according to the original dimension. Random mapping has been successful [2, 56, 3, 57, 37, 58] to reduce the size of features when encoding the word category map, with small loss of information.

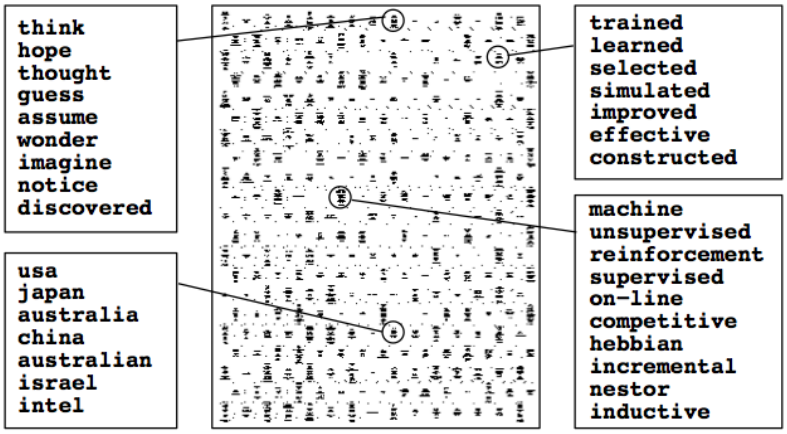


Figure 2.8: Case study in [1]: A Usenet Newsgroup.

The case study in [1], organised approximately $12 * 10^5$ words into 315 word categories. In figure 2.8 we can visualise some word categories of this case study. The result presents a very large dimension reduction, and the categories presented in the figure seem good samples. However, building categories can result in a loss of semantic sharpness and ambiguity. For instance, the category composed by the set of countries ("usa", "japan","australia", "china", "israel") represent those elements as one, so in a further indexing these words are indexed as the same. For this dissertation we aim to solve ambiguity by introducing multi-word expressions (MWEs) and using a different dimension reduction, so that there is no loss of semantic sharpness.

2.3.2 Document Map

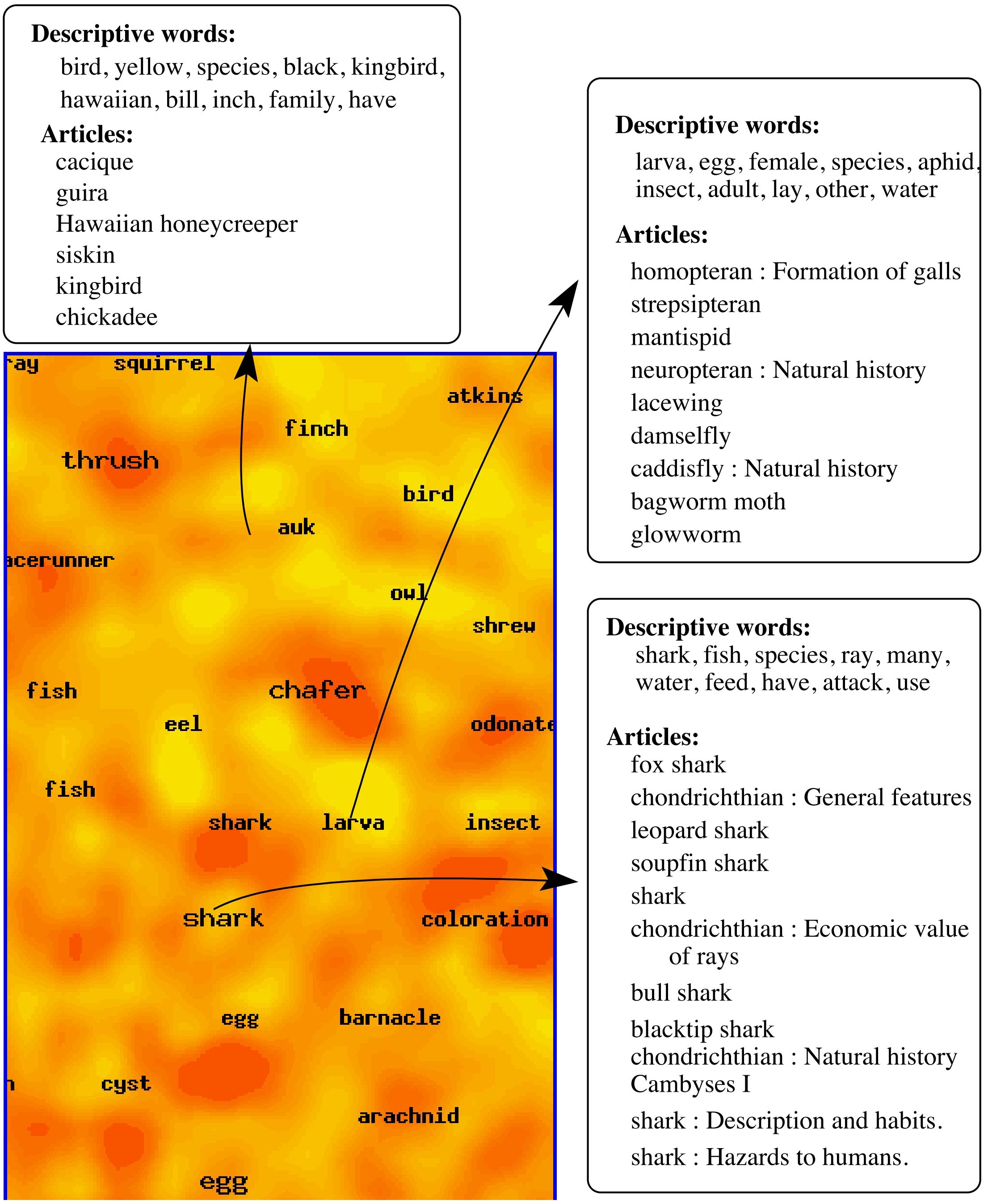
Documents are encoded by locating their word histograms in the word category map. For this encoding, each document is represented by the VSM, however, instead of single words the components are word categories. The category weight can be set by the inverse frequency of occurrence or some other traditional method [59]. In some WEBSOM case studies [57, 60, 53, 3], the entropy-based weight (section 2.2.2.2) is used to index the category values. Once the encoding is completed, the document map is formed with the SOM algorithm using the categories as *fingerprints* of the documents.

The WEBSOM method has successfully formed document map for different types of free-text collections [1, 2, 52, 53, 56, 3, 57, 37]. Some of these publications are implemented at [61] in a demo version to browse and explore the document map. The largest document map [3] had approximately 7 million patent abstracts written in English, where the average length of each text was around 132 words. The document map took around six weeks to be computed, even with the reduction methods applied. The result had the documents ordered meaningfully on a document map according to their contents. For computational reasons, the present dissertation aims to perform a document map for smaller collections, as was done in [1, 52, 37].

The case study in [2] organised a collection of 115 000 articles from the Encyclopaedia Britannica. The *corpus* was preprocessed to remove unnecessary data, which resulted in an average length of 490 word per document and a vocabulary of 39 000 words. In this particular study, the word category map was ignored, having the document encoded as a random projection of word histograms. The random projection consisted in a dimension of 1000 features and the number of ones in each column of the sparse random projection matrix was three. In figure 2.9 we can visualise a sample of the document map. Although the word category map step was ignored, the result continues to organise documents according to their contents. This case study result exemplifies one of the motivating factors for this thesis that is to use the WEBSOM method without using the word category map.

2.3.3 Browsing The Document Map

Browsing the document map is presented to the user as a series of HTML pages that enable the exploration of the map. When clicking on a point on the map, it links to a



Descriptive words:
bird, yellow, species, black, kingbird, hawaiian, bill, inch, family, have

Articles:
cacique
guira
Hawaiian honeycreeper
siskin
kingbird
chickadee

Descriptive words:
larva, egg, female, species, aphid, insect, adult, lay, other, water

Articles:
homopteran : Formation of galls
strepsipteran
mantispid
neuropteran : Natural history
lacewing
damselfly
caddisfly : Natural history
bagworm moth
glowworm

Descriptive words:
shark, fish, species, ray, many, water, feed, have, attack, use

Articles:
fox shark
chondrichthian : General features
leopard shark
soupfin shark
shark
chondrichthian : Economic value of rays
bull shark
blacktip shark
chondrichthian : Natural history
Cambyses I
shark : Description and habits.
shark : Hazards to humans.

Figure 2.9: Case study in [2]

document database [52]. If the map is large, subsets of it can first be viewed by zooming [3], as shown in figure 2.9. To provide guidance for the exploration, an automatic labelling method (section 2.3.3.1) has been applied to rank keywords for different map regions [62]. Figures 2.9 and 2.10 are examples using the automatic labelling. In addition, some WEBSOM examples are available for browsing at [61].

2.3.3.1 Automatic labelling

In order to convey an intuitive manner for browsing one needs descriptive "landmarks" to be assigned to the document map regions where particulars topics are discussed. The automatic labelling [62] is based on two statistical properties: the words occurrence in the cluster; and the words occurrence in the collection. The goodness of a word w in a cluster j is measure by

$$G(w,j) = F(w,j)\frac{F(w,j)}{\sum_i F(w,i)}\ , \tag{2.18}$$

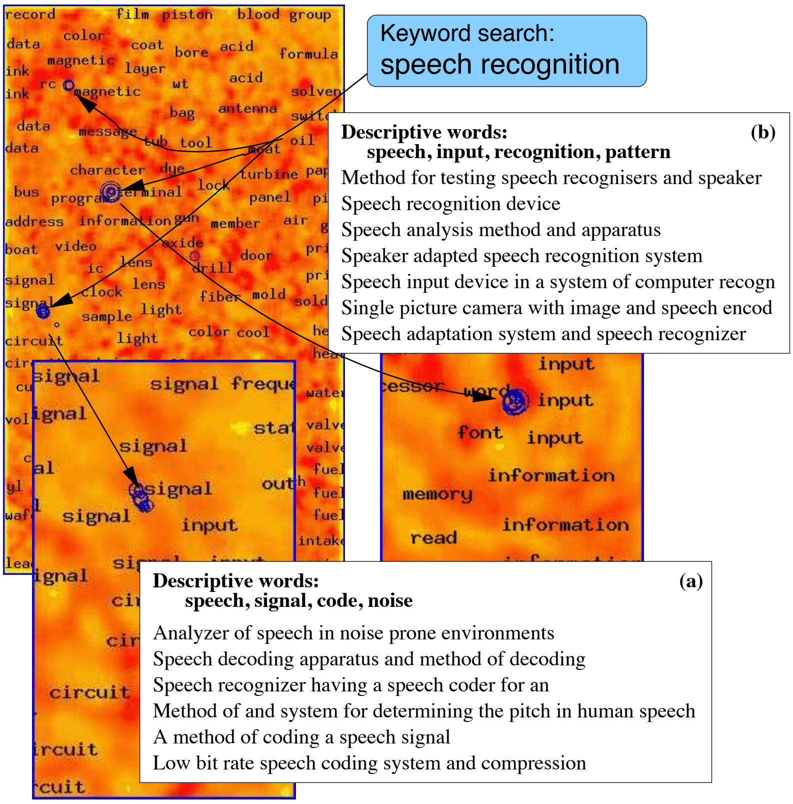
where $F(w,c)$ stands for the frequency of the word w in a cluster c. Based on the previous equation, the words which occur in the cluster are ranked, and then the top one is denoted as the representative of the cluster. However, the top words are still considered and ranked, as we can visualise in figure 2.9. The labelling is an useful tool to provide an overview about the most relevant keywords that occur in a cluster or region of the map. In addition, if the keywords have a strong meaning they can give a perception of the main topics.

2.3.3.2 Content Addressable Search

The content addressable search is a method that processes the content text into a document vector in the same manner as done in the data preparation. The resulting vector is then compared with the model vectors of all map units, and the best-matching units are computed and then saved. The output ranking is an ordered list about the most similar items between the content text and model vectors. This method is often used to visualise where new textual information will be located in the document map.

2.3.3.3 Keyword Search

The keyword search is also a good method for searching the document map. After building the map for each word the map units that contain the word are indexed. Given a search description, the matching units are found from the index and the best matches are saved. The ranking is then computed according to the best matches. In figure 2.10 we can visualise keyword search for "speech recognition", where the matching document areas are marked by circles. When searching in the marked areas, the matching units are ranked according to their resemblance to the query.



Keyword search:
speech recognition

Descriptive words: **(b)**
speech, input, recognition, pattern
Method for testing speech recognisers and speaker
Speech recognition device
Speech analysis method and apparatus
Speaker adapted speech recognition system
Speech input device in a system of computer recogn
Single picture camera with image and speech encod
Speech adaptation system and speech recognizer

Descriptive words: **(a)**
speech, signal, code, noise
Analyzer of speech in noise prone environments
Speech decoding apparatus and method of decoding
Speech recognizer having a speech coder for an
Method of and system for determining the pitch in human speech
A method of coding a speech signal
Low bit rate speech coding system and compression

Figure 2.10: Case study in [3]

2.4 Keyword Extractor

For this thesis study we intend to characterise documents by multi-word expressions (MWEs), instead of single words. The reason relies on the ambiguity of single words which can be solve when using MWEs. For instance, [63] demonstrates that the lemma *world* has nine different meanings and *record* has fourteen, while the MWE world record has only one. In addition, when using indexed relevant MWEs, the system accuracy improves [64]. Section 2.4.1 gives a brief survey about MWE extractors and section 2.4.2 presents a detailed description about the MWEs extractor used in this dissertation.

2.4.1 Multi-Word Expressions Extractors

Regarding multi-word expression extractors, there are *linguistic* and *statistical* approaches. Linguistic approaches like [65, 66, 67, 68, 69, 70, 71] use syntactical filters, such as *Noun-Noun*, *Adjective-Noun*, *Verb-Noun*, etc., to identify or extract MWEs. Since the textual data has to be morphosyntactically tagged, this requirement imposes a language dependency on linguistic approaches. Not all languages have high quality taggers and parsers available, especially when languages are unknown. Furthermore, the MWEs relevance is not assured by morphosyntactic patterns. For example, "triangle angle" and "greenhouse effect" share the *Noun-Noun* pattern, however only the second one can be considered relevant.

Statistical approaches are usually based on the condition that many of the words of a MWEs are *glued*. For example, there is a high probability that in texts, after the word *Barack*, appears the word *Obama*, and that before *Obama* appears the word *Barack*. Several statistical measures, such as *Mutual Information* [72], *Coefficient* [73], *Likelihood Ratio* [74], etc., have been used to obtain MWEs. The problem with these measures is that they only extract bigrams (sequences of only two words). However, in [75] proposes other metrics and an extraction algorithm (section 2.4.2), to obtain relevant MWEs of two or longer words. For this dissertation, a statistical approach is used to obtain MWEs according to their relevance in the collection. Besides, statistical methods are independent from the language which is useful to test this study approach on different languages, in specific Portuguese and English documents.

2.4.2 LocalMaxs

Three tools working together, are used for extracting MWEs from any corpus, the LocalMaxs algorithm, the Symmetric Conditional Probability (SCP) statistical measure and the Fair Dispersion Point Normalization (FDPN).

For a simple explanation, let us consider that a n-gram is a string of n consecutive words. For example the word *president* is an 1-gram; the string *President of the Republic* is a 4-gram. LocalMaxs is based on the idea that each n-gram has a kind of *glue* or cohesion sticking the words together within the n-gram. Different n-grams usually have different cohesion values. One can intuitively accept that there is a strong cohesion within the n-gram (*Barack Obama*) i.e. between the words *Barack* and *Obama*. However, one cannot say that there is a strong cohesion within the n-gram (*or uninterrupted*) or within the (*of two*). So, the $SCP(.)$ cohesion value of a generic bigram $(x\ y)$ is obtained by

$$SCP(x\ y) = p(x|y)\,.\,p(y|x) = \frac{p(x\ y)}{p(y)}\,.\,\frac{p(x\ y)}{p(x)} = \frac{p(x\ y)^2}{p(x)\,.\,p(y)} \tag{2.19}$$

where $p(x\ y)$, $p(x)$ and $p(y)$ are the probabilities of occurrence of bigram $(x\ y)$ and unigrams x and y in the corpus; $p(x|y)$ stands for the conditional probability of occurrence of x in the first (left) position of a bigram in the text, given that y appears in the second (right) position of the same bigram. Similarly $p(y|x)$ stands for the probability of occurrence of y in the second (right) position of a bigram, given that x appears in the first (left) position of the same bigram.

However, in order to measure the cohesion value of each n-gram of any size in the corpus, the FDPN concept is applied to the $SCP(.)$ measure and a new cohesion measure, $SCP\_f(.)$, is obtained.

$$SCP\_f(w_1 \dots w_n) = \frac{p(w_1 \dots w_n)^2}{\frac{1}{n-1}\sum_{i=1}^{n-1} p(w_1 \dots w_i).p(w_{i+1} \dots w_n)} \tag{2.20}$$

where $p(w_1 \dots w_n)$ is the probability of the n-gram $w_1 \dots w_n$ in the corpus. So, any n-gram of any length is "transformed" in a pseudo-bigram that reflects the *average cohesion* between each two adjacent contiguous sub-n-gram of the original n-gram. Now it is possible to compare cohesions from n-grams of different sizes.

LocalMaxs Algorithm

LocalMaxs [75, 76] is a language-independent algorithm to extract cohesive n-grams of text elements (words, tags or characters).

Let $W = w_1 \dots w_n$ be an n-gram and $g(.)$ a cohesion generic function. And let: $\Omega_{n-1}(W)$ be the set of $g(.)$ values for all contiguous $(n-1)$-grams contained in the n-gram W; $\Omega_{n+1}(W)$ be the set of $g(.)$ values for all contiguous $(n{+}1)$-grams which contain the n-gram W, and let $len(W)$ be the length (number of elements) of n-gram W. We say that

W is a MWE if and only if,

$$\begin{aligned} &\text{for } \forall x \in \Omega_{n-1}(W), \forall y \in \Omega_{n+1}(W) \\ &\qquad (len(W) = 2 \, \wedge \, g(W) > y) \quad \vee \\ &\qquad (len(W) > 2 \, \wedge \, g(W) > \frac{x+y}{2}) \ . \end{aligned}$$

Then, for n-grams with $n \geq 3$, LocalMaxs algorithm elects every n-gram whose cohesion value is greater than the average of two maxima: the greatest cohesion value found in the contiguous $(n{-}1)$-grams contained in the n-gram, and the greatest cohesion found in the contiguous $(n{+}1)$-grams containing the n-gram.

$SCP\_f(.)$ cohesion function were used as $g(.)$ in the LocalMaxs algorithm.

3

Proposed Approach

3.1 Architecture

In this study we propose a method to characterise documents and then extract new information based on the characterised dada. The proposed architecture is presented in figure 3.1. The method is designated as the RSOM approach.

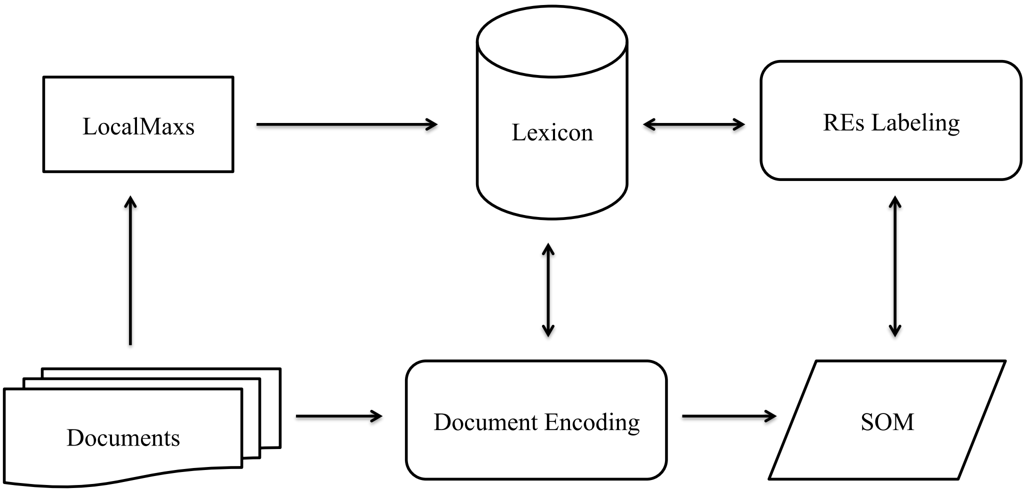


Figure 3.1: Proposed Architecture.

The first step is to build the Lexicon that is composed by multi-word expressions (MWEs) extracted from documents using the LocalMaxs algorithm. These MWEs are then used in the Document Characterisation/Encoding component as document features to build a structure that represents each document from the collection. There are other approaches that set MWEs occurrence or frequency to characterise documents, but these are computationally heavy for the selected algorithms in the further component. So, the structure produced in the Document Characterisation component includes a reduction of the number of attributes to characterise the documents. This component is explained in detail in section 3.2. The previous components form the necessary steps to prepare the data for the SOM component. In this component is projected a structured map that is organised according to the document features. The Self-Organised Map (SOM) is used in this component, because this method provides visual tools for the information extraction. However, there is some information that cannot be visualised just using SOM. So, the REs Labelling component is an extension that provides a set of techniques which interact with the Lexicon and SOM component.

3.2 Document Information Retrieval

3.2.1 Document Characterisation

In this study documents are characterised by features. Most approaches use a great number of features, usually based on the entire lexicon available in the *corpus*, which become computationally heavy. So, instead of using all elements of the vocabulary in the corpus, a different and reduced set of features was calculated for each document. Thus, by using the MWEs it was possible to calculated a similarity matrix between each pair of documents. This way, each document is characterised by a new vector where each position reflects the similarity between this document and another document of the *corpus*. This corresponds to a strong feature reduction and there is no loss of information since all MWEs enter in the calculation of the similarity matrix. As an example, from a *corpus* with 148 documents used in this dissertation, 5818 MWEs were extracted. The usual form would use all expressions in the vocabulary to represent each document, which in this case would correspond to 5818 features. Instead of this number, there is a compression of 5818 to 148 (the number of documents) and this reduced number of attributes is used to characterise the same documents. In figure 3.2, the left matrix is replaced by the right matrix,

$$\begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,vlen} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,vlen} \\ \vdots & \vdots & \ddots & \vdots \\ w_{dlen,1} & w_{dlen,2} & \cdots & w_{dlen,vlen} \end{pmatrix} \Rightarrow \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,dlen} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,dlen} \\ \vdots & \vdots & \ddots & \vdots \\ s_{dlen,1} & s_{dlen,2} & \cdots & s_{dlen,dlen} \end{pmatrix}$$

Figure 3.2: Reduction example.

where $w_{k,i}$ means the probability of the attribute i in document k and $s_{k,q}$ stands for the similarity between documents k and q, which is based in Pearson correlation, equation 3.1. The $vlen$ denotes the length of the vocabulary and $dlen$ the number of documents.

$$s_{k,q} = \frac{cov(k,q)}{\sqrt{cov(k,k)} * \sqrt{cov(q,q)}} \tag{3.1}$$

$$cov(k,q) = \frac{1}{vlen-1} \sum_{i=1}^{i=vlen} (w_{k,i} - w_{k,.}) * (w_{q,i} - w_{q,.}) \tag{3.2}$$

where $w_{k,.}$ means the average value of the attributes for document k which is given by

$$w_{k,.} = \frac{1}{vlen} \sum_{i=1}^{i=vlen} w_{k,i} \quad . \tag{3.3}$$

The value of $w_{k,.}$ tends to be very low because a document k has only a small set of all MWEs in the corpus and this sum is divided by the number of all elements in the vocabulary.

In Pearson correlation, similarities range from -1 to +1, so if $s_{k,q}$ is close to 0 it means that the similarity between documents is weak; if it is close to +1, the similarity between k and q is strong; if negative, it means that documents are dissimilar. This scale is the same in all cases, which allows comparison of different correlations. In addition, by looking at $(w_{k,i} - w_{k,.}) * (w_{q,i} - w_{q,.})$ in equation 3.2, which is the contribution of MWE i in the result, each product either enhances or penalises the similarity between both documents. The different types of contributions are:

1. $w_{k,i} > 0 \wedge w_{q,i} > 0$: When MWE i occurs in both documents, the subtractions $(w_{k,i} - w_{k,.})$ and $(w_{q,i} - w_{q,.})$ are positive, because $w_{k,i}$ and $w_{q,i}$ are greater than the corresponding average values $w_{k,.}$ and $w_{q,.}$. Being both positive, this product enhances the similarity between documents k and q. The impact of this contribution varies according to the probability of MWE i in both documents, where high probabilities have a greater weight in the result.

2. $(w_{k,i} > 0 \wedge w_{q,i} = 0) \vee (w_{k,i} = 0 \wedge w_{q,i} > 0)$: Since MWE i occurs in only one of the documents, the product $(w_{k,i} - w_{k,.}) * (w_{q,i} - w_{q,.})$ will be negative because one of the subtractions is negative and the other is positive. So, the similarity is penalised when the MWE occurs in only one of the documents.

3. $w_{k,i} = 0 \wedge w_{q,i} = 0$: If a MWE does not occur in either documents, both subtractions $w_{k,i} - w_{k,.}$ and $w_{q,i} - w_{q,.}$ are negative. So, the product $(w_{k,i} - w_{k,.}) * (w_{q,i} - w_{q,.})$ is simplified to $w_{k,.} * w_{q,.}$, which is a weak positive value. This coherent to the idea that two documents must be considered strongly similar when they have MWEs in common and weakly similar when they share the absence of MWEs.

In this work, we need a tool to measure the individual contribution of each MWE in the similarity between documents k and q. As we will see, this is useful to highlight what are the MWEs responsible for a similarity result. So, based on equation 3.1, the individual contribution is measured by the following metric:

$$score(MWE, k, q) = \frac{(w_{k,MWE} - w_{k,.}) * (w_{q,MWE} - w_{q,.})}{\sqrt{cov(k,k)} * \sqrt{cov(q,q)}} \tag{3.4}$$

As said before, MWEs are extracted by LocalMaxs that retrieves many relevant expressions. However, this algorithm also extracts terms that are not informative and shouldn't be considered as good connectors between documents. In order to connect documents based on relevant expressions, MWEs are weighted in a way that strong terms will provide greater similarities and weak term tend to be ignored. Therefore to determine the weight attribute, the probability of occurrence of MWE i in a document k ($w_{k,i}$) is used as a starting point. Thus, other variants of $w_{k,i}$ are applied to obtain the best possible similarity of results. The next section presents a study about metrics to weight MWEs.

3.2.2 Metric Improvements

The **U* metric** is a variant of $w_{k,i}$ meant to grow similarity scores. In some text collections, correlation results are low but correspond to authentic similarities due to relevant expressions that are common in the documents. In this case it is useful to emphasise the results, so that a high score reflects a strong similarity between both documents and weak correlations are kept weak. Besides, it becomes easier to distinguish high from low correlation results. The metric is given by

$$w_{k,i} = u^*_{k,i} = p_{k,i} * p_{k,i}$$

where $p_{k,i}$ stands for the probability of the MWE i in the document k. Since the probability value ranges from 0 to 1, all scores decrease with the product. However, the smaller the value the greater the decrement. For example if MWE i has the probability of $0,01$ in document k ($p_{k,i}$), the product is 0,0001 so the decrement is quite high. On the other hand, if $p_{k,i}$ is $0,1$, the score is $0,01$ which is a much smaller decrement than before. Figure 3.3 presents two similarity matrices showing the differences between applying simple probability $w_{k,i} = p_{k,i}$ and the probability weighted by itself $w_{k,i} = u^*_{k,i} = p_{k,i} * p_{k,i}$. As we can see, significant similarities are highlighted.

$$\begin{pmatrix} 1 & 0.545 & 0.244 & 0.243 & 0.383 \\ 0.545 & 1 & 0.307 & 0.376 & 0.487 \\ 0.244 & 0.307 & 1 & 0.16 & 0.259 \\ 0.243 & 0.376 & 0.16 & 1 & 0.418 \\ 0.383 & 0.487 & 0.259 & 0.418 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0.728 & 0.482 & 0.446 & 0.509 \\ 0.728 & 1 & 0.248 & 0.688 & 0.62 \\ 0.482 & 0.248 & 1 & 0.099 & 0.222 \\ 0.446 & 0.688 & 0.099 & 1 & 0.6 \\ 0.509 & 0.62 & 0.222 & 0.6 & 1 \end{pmatrix}$$

Figure 3.3: Example of the differences between applying the simple probability and the probability weighted by itself.

Although the application of $u^*_{k,i}$ has improved the authentic correlations, it provoked an unwanted side effect: since the MWE extractor is not perfect, it also extracts MWEs such as "por cento" or "for example". These types of terms occurs with high frequencies, which gets a unwanted importance in the calculation of $w_{k,i} = u^*_{k,i} = p_{k,i} * p_{k,i}$. This effect causes wrong similarities. In order to eliminate this effect, other metrics were developed.

The **UMin Metric** assigns weights based on the size of the words at the beginning or end of MWEs. In this metric it is intended to penalise MWEs that start or end with stop-words, because these elements are considered to be weak terms. However when a stop-word occurs between words, it may correspond to a valid MWE, so for this reason only words in the edges of MWEs are considered. The metric is given by

$$w_{k,i} = UMin_{k,i} = p_{k,i} * min(len(i_1), len(i_n))$$

where i_1 stands for the first word in a MWE and i_n the last one. The shortest word between i_1 and i_n sets the weight ($min(len(i_1), len(i_n))$), which is then multiplied by the probability $p_{k,i}$. The product $p_{k,i} * min(len(i_1), len(i_n))$ should reflect better results when MWEs have relevant words in the edges because these terms usually are longer than stop-words. Figure 3.4 illustrates how the results are affected by using *UMin* metric. The correlation values in the left matrix are calculated having $w_{k,i} = p_{k,i}$ and the right matrix is calculated with $w_{k,i} = UMin_{k,i}$.

$$\begin{pmatrix} 1 & 0.148 & 0.172 & 0.305 & 0.214 \\ 0.148 & 1 & 0.155 & 0.159 & 0.206 \\ 0.172 & 0.155 & 1 & 0.185 & 0.685 \\ 0.305 & 0.159 & 0.185 & 1 & 0.206 \\ 0.214 & 0.206 & 0.685 & 0.206 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0.009 & 0.008 & 0.232 & 0.034 \\ 0.009 & 1 & 0.004 & 0.115 & 0.122 \\ 0.008 & 0.004 & 1 & 0.035 & 0.562 \\ 0.232 & 0.115 & 0.035 & 1 & 0.069 \\ 0.034 & 0.122 & 0.562 & 0.069 & 1 \end{pmatrix}$$

Figure 3.4: Example of the differences between applying the simple probability and the probability weighted by UMin

The similarities become more authentic because strong MWEs have more impact in the correlation result. In tables 3.1 and 3.2 are listed the MWEs and respective $score(.,.,.)$ with a significant contribution to the correlation between two documents. The MWE $score(.,.,.)$ is calculated by equation 3.4, where $w_{k,i} = p_{k,i}$ in table 3.1 and $w_{k,i} = UMin_{k,i}$ in table 3.2. The number of MWEs in table 3.2 is lower than the number in table 3.1, because some elements had their $score(.,.,.)$ reduced to a small value and were no longer considered as a significative contribution to the result. For example the wrong MWE "que a" is reduced from 277,2 to 15,5 because of the word "a" that penalises the term. On the other hand the MWE "cidade de tauranga" gets enhanced from 277,2 to 568,9 because it has long words that provide a good $score(.,.,.)$.

Although this metric has disadvantages: the variation range of the words length is not enough to penalise the high frequency of the wrong MWEs. For example the correlation between documents 1 and 4 had weak terms like "com a" and "que o" that were still considered because they had a very high frequency. Besides, some small words are still meaningful and the MWEs containing them should not be penalised; on the other hand, some stop words have a similar size as some meaningful words and therefore will be equally weighted, which is not correct; ex: "besides" vs "mayor"; "Besides" is a stop word, but "mayor" is meaningful. In order to improve the previous result a new metric was developed with a different weight attribution.

The **Invert Frequency Metric** assigns weights based on the frequency of the words at

$p_{k,i}$	
$score(.,.,.)$	MWE
1668,8	nova zelândia
555,3	recife astrolabe
277,2	cerca de
277,2	que a
277,2	22 quilómetros ao largo da cidade
277,2	ilha do norte da nova zelândia
277,2	largo da cidade de tauranga
277,2	manhã no recife astrolabe
277,2	bandeira da libéria
277,2	cidade de tauranga
277,2	ilha do norte
277,2	manhã no recife
277,2	navio de carga

Table 3.1: Top MWEs between documents 3 and 5 having $w_{k,i} = p_{k,i}$.

$UMin_{k,i}$	
$score(.,.,.)$	MWE
518,8	nova zelândia
1138,8	recife astrolabe
790,5	bandeira da libéria
568,9	cidade de tauranga
568,9	largo da cidade de tauranga
394,8	cerca de 22 quilómetros
394,8	manhã no recife astrolabe
394,8	navio de carga

Table 3.2: Top MWEs between documents 3 and 5 having $w_{k,i} = UMin_{k,i}$.

the beginning or end of MWEs. This metric is a variant of UMin where it is desired to penalise MWEs that start or end with stop-words. Besides stop-words, other terms with a high frequency should be weighted in the same manner, so that strong MWEs are composed by expression with relevant words in the *corpus*. For example the word "petrol" may be a relevant in collection that has different types of news. However if it occurs in a collection that only has petrol news, this word does not have much importance. The metric is given by

$$w_{k,i} = UInv_{k,i} = p_{k,i} * min(invfreq(i_1), invfreq(i_n)) \tag{3.5}$$

$$invfreq(w) = \frac{1}{f(w,.)} \tag{3.6}$$

where i_1 stands for the first word in a MWE and i_n the last one. The $f(w,.)$ means the frequency of the word w in the *corpus* and $invfreq(w)$ the inverse of this frequency. The weight assigned is the lowest value between $invfreq(i_1)$ and $invfreq(i_n)$, which is then multiplied by the probability $p_{k,i}$. The product $p_{k,i}*min(invfreq(i_1), invfreq(i_n))$ should reflect better results when the frequency of the words at the beginning or end is not very high and is able to penalise weak MWEs even if they have a high probability. Figure 3.5 presents two similarity matrices showing the differences between applying simple probability $w_{k,i} = p_{k,i}$ (left matrix) and the probability weighted by $w_{k,i} = UInv_{k,i} = p_{k,i} * min(invfreq(i_1), invfreq(i_n))$ (right matrix). As it can be seen, similarity results are very different.

$$\begin{pmatrix} 1 & 0.148 & 0.172 & 0.305 & 0.214 \\ 0.148 & 1 & 0.155 & 0.159 & 0.206 \\ 0.172 & 0.155 & 1 & 0.185 & 0.685 \\ 0.305 & 0.159 & 0.185 & 1 & 0.206 \\ 0.214 & 0.206 & 0.685 & 0.206 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 0.508 & 0 \\ 0 & 1 & 0 & 0.014 & 0.01 \\ 0 & 0 & 1 & 0.018 & 0.919 \\ 0.508 & 0.014 & 0.018 & 1 & 0.002 \\ 0 & 0.01 & 0.919 & 0.002 & 1 \end{pmatrix}$$

Figure 3.5: Reduction example.

Correlations that are connected by false MWEs have a very low value and correlations connected with strong MWEs have a high $score(.,.,.)$. In table 3.3, the top MWEs of the correlation between documents 3 and 5.The list is composed by valid terms that correspond to an authentic correlation. Valid terms like "ilha do norte da nova zelândia" (northern island of New Zealand), that were previously discarded are now considered because these terms are measured according to their real relevance in the *corpus*. False MWEs like "cerda de" and "que a" are penalised and no longer are considered because their $score(.,.,.)$ is reduced from 277.2 to 0.0001. On the other hand the MWE "nova zelândia" (New Zealand) that is the name of a country was no longer considered because it has the word "nova" (new) that is frequent in the collection. Although some MWEs may be improperly penalised, this metric proved to be efficient to measure the weight of the majority of MWEs responsible for the similarity between documents.

$UInv_{k,i}$	
score	MWE
11888,8	recife astrolabe
1173,2	bandeira da libéria
422,1	ilha do norte da nova zelândia
98,6	largo da cidade de tauranga
56,3	navio de carga
46,7	22 quilómetros ao largo
31,1	manhã no recife astrolabe

Table 3.3: Top MWEs between documents 3 and 5 having $w_{k,i} = UInv_{k,i}$.

Using this metric, it is possible to select a smaller set of MWEs that are really responsible for any similarity. This happens because of the scale produced by this measure is greater and there is a strong decrease from the first scored MWEs and the rest of them. Besides, the global ranking of the MWEs is more correct for this metric. In other words, the global weight of these score values are concentrated in the first MWEs. Thus by using only a few MWEs, this offers an advantage because it is possible to capture the *core semantic reason* of any similarity between two documents.

For example, in table 3.3, due to their most relevant scores, MWEs "recife astrolabe" (Astrolabe Reef), "bandeira da libéria" (Flag of Liberia) and "ilha do norte da nova zelândia" may be selected as the *core semantic reason* for this particular correlation/similarity.

3.3 Relevant SOM

The similarity matrix is used in the Self-Organised Map (SOM) to produce a *neural network*/map. This network is organised according to the document features, having similar documents located in same area of the map. For this work a package is used that implements the Self-Organised Maps and other tools to analyse the *neural network*. However, the similarity matrix does not provide information related to documents MWEs that can be used in SOM. So, in this section methods are proposed to integrate MWEs with SOM.

3.3.1 Documents Keyword ranking

Documents on this approach are associated to a set of MWEs that compress the main topic of the text. These MWEs cannot be visualised with the U-matrix or used with the tools available in the SOM package. This happens because the features in the input data for the SOM have no information related to MWEs. The first step to integrate MWEs with SOM is a method to extract the top terms between documents. Based on equation 3.4 we propose a method to measure the top MWEs from a set of documents:

$$score(MWE,\mathcal{D}) = freq(MWE,\mathcal{D}) * \sum_{d_i \in \mathcal{D}} \sum_{d_k \in \mathcal{D}, d_k \neq d_i} score(MWE, d_i, d_k) \ , \tag{3.7}$$

where $\mathcal{D}$ is a set of documents and MWE is a multi-word unit that occurs in $\mathcal{D}$. For all $MWEs$ in $\mathcal{D}$ is calculated a $score(.,.)$, that is obtained by the product of the frequency of the MWE in $\mathcal{D}$ and the sum of all document pairs $score(.,.,.)$ in which the MWE occurs. The top MWEs are those that have higher scores. So, the MWE that has the higher value is the most influential from the documents in $\mathcal{D}$. This methodology is also used to obtain the top $MWEs$ from a neurone or a cluster (group of neurones). To compute the ranking it is only necessary to specify the documents contained in the neurone or neurones in the case of a cluster.

The following example presents how this methodology is used for this work: The U-Matrix in figure 3.6 is a small map with petrol news. The numbered hexagons represents the id of a neurone with documents, when a neurone has no documents is labeled by an arbitrary character (in this case is set with "-"). These ids are a glue to identify the neurones and their location in the map, so that after performing the ranking there's a perception of the MWEs and their contribution in the *neural network*. Table 3.4 presents the ranking for neurone 31 of the U-matrix presented in figure 3.6. The result gives a idea about the main topics of the neurone, where "light sweet" highlights as the most influent of the ranking. According to the U-Matrix (figure 3.6), it can be seen that neurone 31 and 32 are connected. Therefore, when computing the ranking for neurone 32 it is verified that the MWE "light sweet" is also influent in neurone 32. Besides this one, "nova

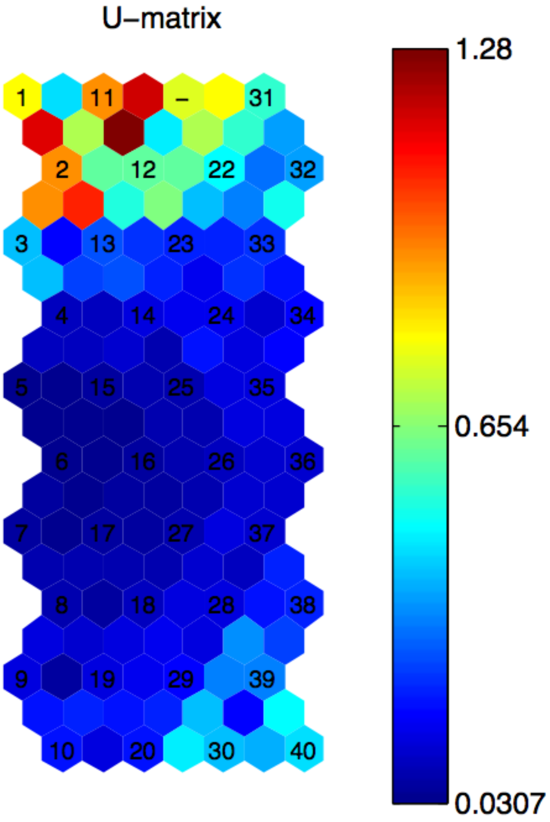


Figure 3.6: A small *neural network* with petrol news.

score	MWE
12560262,2	light sweet
216891,2	nova iorque
158466,2	entrega em dezembro
90097,4	barril de brent
36276,9	sessão anterior
14544,2	dezembro fechou

Table 3.4: Top MWEs for neurone 31.

iorque" (New York) also rises as one of the top MWEs. Although, the influence of the score is different from one neurone to the other, they share a semantic resemblance and the connection is related to the MWEs "light sweet" and "nova iorque". There are other MWEs in common, but their influence in those neurones was very low, so they were not considered as relevant.

score	MWE
1489951,1	york mercantile
372279,9	new york mercantile exchange
163320,5	light sweet
1412,9	nova iorque
1274,4	chakib khelil
176,9	semana passada

Table 3.5: Top MWEs for neurone 32

This study presented how the methodology can provide information about the impact of MWEs in a neurone, that could not be obtained before. This study also demonstrated that some neurones share the same top MWEs, which related neurones based on expressions. Although it is possible to achieve the previous information, this analysis is exhaustive because in order to obtain the necessary information it is necessary to compute neurone by neurone. So, in the next section we propose a methodology to visualise the impact of a MWE in the *neural network*.

3.3.2 Keyword Component Plane

Features based on the similarity between documents are useful in the context of SOM because they add attributes that are used as semantic properties. As an example, two documents may not have a high correlation between them, but can be related based on similarities with other documents. As has been explained before (section 3.2.1), similarities between documents are measures by the MWEs, and it is through these that one is able to identify some semantic relations. To analyse the individual impact of a MWE in the *neural network* we propose a method similar to the component planes. The structure is the same (as the component planes), however the neurones are set in a different manner:

$$CPK(i, \mathcal{KD}) = \sum_{d \in \mathcal{KD}} n_{i,d} * contain(i, d) \ , \tag{3.8}$$

where i represents the neurone id and $\mathcal{KD}$ stands for all documents that contain the MWE that is being analysed. The $n_{i,d}$ means the codebook value for neurone i and document d and it is only added to the sum if the document belongs to the neurone. So, if the neurone i contains the document d the value of $contain(i, d)$ is 1, otherwise is 0. The value set for element i is given by the sum of all codebook values that belong to neurone i and have the expression MWE. In figure 3.7 we project the impact of the MWE "light sweet" (left projection) and the U-Matrix (right projection) that is being analysed. The colour set for each hexagon represents the impact of the MWE in the neurone, in which this scale is displayed on the right of the U-Matrix. As it can be seen, a dark blue means that the value is low and as brighter the colour becomes the greater is the value. Thus, as the value increases it means that the MWE has more impact in the neurone.

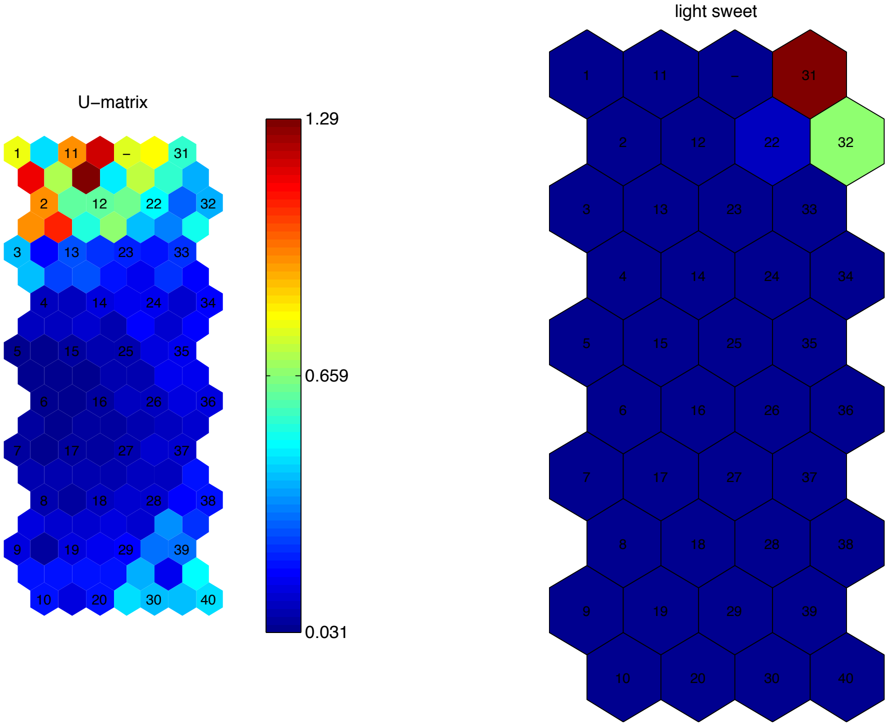


Figure 3.7: Example of MWE "light sweet" component plane.

The projected MWE shows that the expression is influential on neurone 31 and 32 and has a small influence on neurone 22. This result shows that the MWE "light sweet" has some influence in that area of the map and is one of the reasons that they are close. Although the previous expression may be a reason for those elements to be close, other MWEs can also be influential in that area of the map. For instance, based on the tables 3.4 and 3.5 studied in section 3.3.1, it can be visualised that the MWE "nova iorque" is another expression that those neurones have in common. So, the term "nova iorque" is projected (figure 3.8) to obtain an overview of its relevance in the *neural network*. The result reveals that the MWE is relevant in neurone 31 and 32 and has small influence in other areas of the map. Since there are no more top expressions in common between those neurones (neurones 31 and 32), it is associated to the connection the MWEs "nova iorque" and "light sweet". It can also be concluded that these expressions only have a relevant impact in that area, even if they occur in other areas of the U-Matrix.

Through this methodology it is possible to relate the MWE impact in *neural network* without changing the input data structure used in SOM. Some terms have relevance in only one neurone, while others can have relevance in a group of neurones. This type of characteristic is observed in the MWE component plane without the need to consult

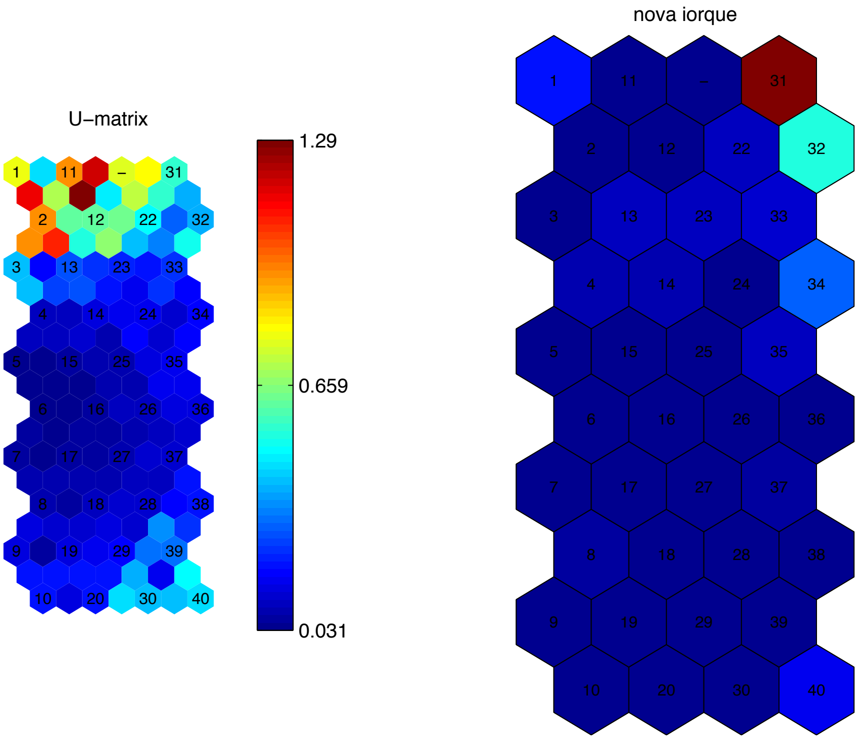


Figure 3.8: Example of MWE "nova iorque" component plane.

neurone by neurone. The Relevant SOM package is composed by methods to rank the top MWEs in a neurone and to project the individual impact of $MWEs$ relative to the *neural network*. Combining both techniques provides a set of tools that improves the perception of the structure and the organisation of the U-Matrix.

4 Results

In this chapter, we analyse the benefits introduced by the proposed approach, namely the RSOM, and study the metrics that were applied in the thesis, in order to select the best REs to characterise documents. The weighted REs are used to correlate documents and form a similarity matrix, so the number of features is reduced. Several parameters are considered to produce the best document map considering different training configurations. Later, neurones of the document map are labeled by REs, since their semantic accuracy describes neurones better than single-words used in the original WEBSOM. Precision and Recall were evaluated for WEBSOM and RSOM concerning the neurones REs labelling and respective document distribution. Finally RSOM conceptual maps are analysed regarding their organisational capabilities.

4.1 Case Study Collections

For this thesis, three different collections were used to examine the proposed approach. The goal is to determine the system behaviour and limitations in relation to different types of textual information, namely size of the texts, number of documents and topic organisation. Table 4.1 summarises the main dimension properties for each collection.

Corpus	Number	Words		Multi-word Expressions	
	Documents	Total	Average	Total	Average
Abstracts	1300	457 000	350	22 000	20
Companies reports	150	155 000	1100	5818	39
News	1000	316 000	315	11 400	11

Table 4.1: Types of documents

The Abstract or *NSF Award Data* is textual collection composed by 129 000 abstracts of NSF-sponsored basic research projects between years 1990 and 2003. The content of the abstract documents is composed by specific expressions related to their research field, and their textual description is very precise, in order to provide a clear perception of the study. Since this *corpus* is large and covers the main research fields in the EUA, a detailed analysis is considered by selecting seven intersecting research fields: *population biology* (PB), *probability* (P), *statistics* (S), *economics* (E), *political science* (PS), *distributed systems and compilers* (DSC), *network infrastructure* (NI) and *computer systems architecture* (CSA). The case study consists of 1300 documents with approximately 350 words per document. The number of extracted MWEs where 22 000 and had around 20 MWEs per text.

The RSOM approach was also tested on different types of documents, such as news and reports. These types of textual documents were selected to intersect them with business activities, however, this analysis is beyond the main goals meant of this thesis. Therefore, these collections are used to compute document maps for further business studies.

The collection composed by *Companies Reports* consists of parts of annual reports released by companies. The content of these documents is about the performance of the company during the year it was released and future objectives which they desire to accomplish on the next years. The *corpus* has a size of 150 documents with approximately a total of 155 000 words and an average of 1050 words per document. For these documents, the LocalMaxs algorithm extracted 5818 MWEs, where each document had associated to it 39 MWEs. Although this collection is small, the textual size is enough for the LocalMaxs to obtain a reasonable set of MWEs for each document.

For this study, a crawler was developed to search and retrieve specific news in an *online journal*. The *News* collection is composed by news retrieved from this crawler. For the *News* collection, different domains were selected, such as: *petrol*; *sports*; *agriculture*; and *medicine*, to compose its structure. One of the goals meant for this collection is to verify if random news on an online journal can be organised into a document map. In addition, since the extracted news have a small size, it is also intended to visualise the quality of the extracted MWEs for smaller documents.

4.2 Relevant Multi-Word Expressions

Relevant Multi-Word Expressions are a collection of keywords that indicate which are the more influential ones in a body of text. In this section a result for: U, UMin and the Invert Frequency metric will be calculated. These variables assign weights in a different manner, therefore each result changes according to how the weight is determined. The results are calculated by the equation introduced in section 3.3.1 and the datasets used to determine the solution are all documents in the selected *corpus*. The text document used was Petrol news.

The results in table 4.2 have all MWE with equal weight, therefore the top keywords

Multi-Word Expression	Total Frequency	Total Documents
por cento	1452	330
que a	622	402
que o	617	395
com a	515	301
o preço	405	221
com o	398	240
que não	179	146
no mercado	174	139
o governo	164	110
acordo com	142	132

Table 4.2: Collection using the U metric

are those that occur more often in the corpus. The top terms using the U metric are not very informative. Many multi-word expressions have a stop-word at the beginning or end, that doesn't add more relevance when combining words. For instance the term "o preço" (that means "the price") has a stop-word "o" and a noun "preço". The stop-word at the beginning of the expression, combining both words, doesn't build a stronger keyword. However, with the term "golfo do méxico" (Gulf of Mexico), there are two nouns "golfo" and "méxico" and one stop-word in the middle. For this example the stop-word is a glue between both nouns, that provide a stronger keyword when analysed together.

Multi-Word Expression	Total Frequency	Total Documents
por cento	1452	330
que a	622	402
o preço	405	221
milhões de euros	175	105
golfo do méxico	150	79
estados unidos	148	111
preços do petróleo	121	102
para entrega	119	92
barril de brent	99	93
referência para portugal	70	69

Table 4.3: Collection using the UMin metric

Table 4.3 shows more informative keywords because of the weighting factor, but some weak terms still remain as relevant. This happens because some have a high frequency in the document that is enough to have a good score. Another problem with this weighted variable is that it also penalises good keywords when they have a short word at the start or end of the expression. Since informative keywords usually don't have a high frequency, they end up penalised.

For example, in the *corpus* there are two documents that have "preços dos combustíveis" and "gasolina sem chumbo 95" in common:

MWE	$p(MWE, doc_{628})$	$p(MWE, doc_{663})$
gasolina sem chumbo 95	0,0093	0,0212
preços dos combustíveis	0,0046	0,0142

Table 4.4: Probability values in respective document.

MWE	$min(MWE_0, MWE_n)$	Score
gasolina sem chumbo 95	2	265
preços dos combustíveis	6	800

Table 4.5: Multi-Word Expressions scores.

The multi-word expressions "gasolina sem chumbo 95" and "preços dos combustíveis" are both informative, however the second keyword has more impact because of its weight. In table 4.4 we have the probability for both terms, although "gasolina sem chumbo 95" has a higher probability the score doesn't enhance because of its small weight. The scores in table 4.5 show how the expressions enhance according to their weight.

Multi-Word Expression	**Total Frequency**	**Total Documents**
golfo do méxico	150	79
maré negra	116	56
nova iorque	127	88
mercados internacionais	83	68
referência para portugal	70	69
gasolina sem chumbo 95	52	35
futuros de londres	42	42
cotação do barril de brent	32	32
arábia saudita	32	24
light sweet	25	24

Table 4.6: Collection using the Invert Frequency Metric

As shown in table 4.6 keywords with a stop word at either end were not considered as relevant. This happened because stop words have a high frequency in a *corpus*, so they were more penalised. These keywords are still considered during the correlation score, however they had a small value added to the final score. Some informative keywords can be more penalised than desired, but this metric provided a good collection of expressions. Similarities that were based on these terms could be considered as correct correlations. This could not be said based on tables 4.2 and 4.3. For this reason the invert frequency metric was used to weight keywords.

4.3 Document Correlation

As it was mantioned in section 3.2.1, document correlation was calculated based on the Relevant Expressions that documents share (or not). Thus, an example of a pair of documents from Company Reports *corpus* with high correlation between them, had the following top common keywords:

1. Peixe Angical
2. lados da fronteira
3. optaram pela adesão àquele

The selected examples were written by the same company, but one was released in 2001 and the other in 2002. The correlation made sense because they talked about the same affair. The paragraphs with the keyword "Peixe Angical" talked about the same matter. The same happened with other paragraphs such as the case that shares the keyword "lados da fronteira"; see table 4.7.

Peixe Angical	Document 2001	*Ainda em 2001, licitámos com êxito as concessões para a construção e exploração das centrais hidroeléctricas de* ***Peixe Angical*** *(452 MW) e Couto Magalhães (150 MW).*
	Document 2002	*Ainda no Brasil e face aos factores de incerteza que mencionei, o Grupo EDP decidiu reprogramar, nesta fase, o desenvolvimento dos projectos das centrais hidroeléctricas de* ***Peixe Angical*** *(452 MW) e Couto Magalhães (150 MW), cujas concessões nos foram atribuídas em 2001*
lados da fronteira	Document 2001	*O Grupo EDP torna-se, assim, na primeira empresa ibérica a deter activos significativos de produção dos dois* ***lados da fronteira****, distribuindo energia eléctrica em Espanha a cerca de meio milhão de clientes.*
	Document 2002	*O Grupo EDP tornou-se, assim, na primeira empresa ibérica a deter activos significativos de produção e distribuição dos dois* ***lados da fronteira****, bem como uma base de clientes de dimensão substancial.*

Table 4.7: Sample connector

These keywords were good connectors in these contexts, as we can see on table 4.7. These are informative keywords. However, in the third case we have "optaram pela adesão àquele" which is weakly informative and, although they are used in the same topic in this case, it could connect other documents not sharing the same topic. In this collection, the LocalMax algorithm provided informative keywords that were able to summarise some paragraphs and link them. The algorithm also extracted incorrect or weak REs, but with the *Invert Frequency metric* we were able to measure the correlation between documents based on the most informative REs, which we consider as keywords. This extraction and weighting was also obtained for the other collections.

Documents		Correlation Value	Multi-Word Expression
Document 2005	*Document 2006*	*0,41*	short list
			parques de estacionamento
			cobrança electrónica
			via verde
			República Checa
Document 2005	Document 2008	0,22	short list
Document 2006	*Document 2008*	*0,44*	segurança rodoviária
			cobrança de portagem
			short list

Table 4.8: Sample correlation

Table 4.8 shows correlation values for pairs of documents from the same company reports *corpus*. Although these values are not very high, they are significant because non significant correlation values are typically lower than 0.1.

Table 4.9 shows correlation values for pairs of short documents, each one having on average 11 REs, but sharing a very informative RE: "Virgílio Constantino". Thus, since documents are short, every RE in them takes a higher importance and then, once there is a common Keyword in every document, it results in high correlations. This shows that this approach is also able to deal with short documents.

Documents		Correlation Value
Document 43	Document 586	0,89
Document 43	Document 1001	0,67
Document 159	Document 43	0,91
Document 159	Document 586	0,96
Document 159	Document 1001	0,67
Document 309	Document 43	0,70
Document 309	Document 159	0,73
Document 309	Document 586	0,71
Document 309	Document 1001	0,5
Document 586	Document 1001	0,67

Table 4.9: Sample correlation

These short documents were comments about petrol and formed a cluster that could be classified as news comments from "Virgílio Constantino". Global results had many clusters that could be classified just by one or a couple of keywords.

We can conclude that LocalMax provided good REs to calculate correlation between documents. The weighting of term improved the results and also showed which keywords were responsible for the correlation value. Although we had good results with the companies reports, determining the expression that classify a group of documents with high values of correlation was not clear because there were many relevant keywords. For documents such as those from the News *corpus* the task to identify was much easier, since the content was smaller and a couple of keywords were enough to relate them.

4.4 Document Map Training

The document map is developed according to the values in the correlation matrix. In other words, documents are characterised by their similarities with the rest of the documents. However, the results can vary according to the training configurations. So, to determine the configurations which produce the best results, the Abstract *corpus* data was trained using different sample sizes of the *corpus* and different types of configurations, such as: *map size* and *training duration*.

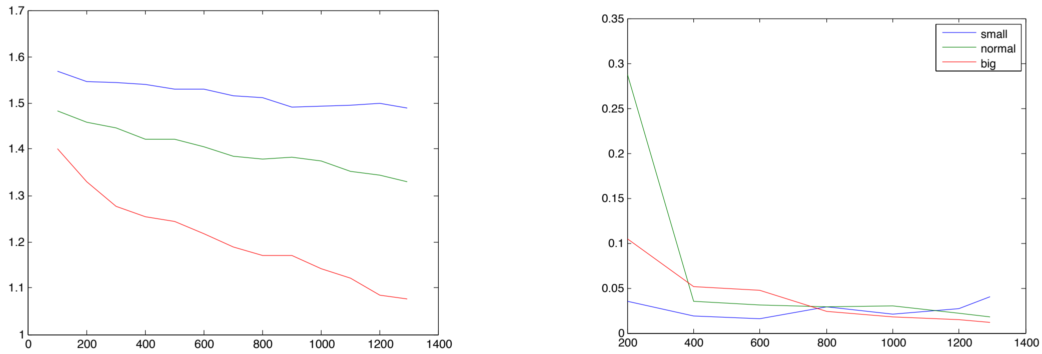
The results were evaluated through two SOM measures, *average quantisation error* and *topographic error*. The average quantisation error (QE) stands for the average distance between input vectors and their BMU, while topographic error (TE) gives the percentage of data vectors for which the BMU and the second-BMU are not neighbouring map units [14]. These measures provide an overview about the training errors, but do not give a perspective about the document distribution on the map. So, in order to obtain an overview of the documents organisation, two accuracy metrics were applied to measure their distribution, that are the *Field Purity* and *Empty Rate*. These metrics are based on the field labelling associated to neurones which is related to the documents field. The first step is a supervised labelling of documents according to their field. For example, for the Abstract collection, documents are labeled according to their *research field*. Then, the next step is to associate each neurone with a field based on the documents which occur in it. When a neurone contains documents with different fields, the field affiliated to it is the one that occurs more often. So, the Field Purity measures the number of documents with the same field as the neurone (equation 4.1).

$$fieldAc(\mathcal{N}, f) = \frac{1}{|\mathcal{N}|} \sum_{d \in \mathcal{N}} sameField(field(d), f) \quad , \tag{4.1}$$

$$sameField(f_i, f_j) = \begin{cases} 1, & f_i = f_j \\ 0, & f_i \neq f_j \end{cases} \tag{4.2}$$

The $\mathcal{N}$ stands for a neurone and $|\mathcal{N}|$ the number of documents in the neurone. In some cases, the map has neurones that do not contain any documents and are labeled as empty. The Empty Rate measures the portion of empty neurones on the map.

The results are computed using *batch algorithm*. In performed experiments this algorithm had a similar output to the *sequential algorithm*, but generated the document map much faster. For instance, for the Abstract collection with the same configurations, the



(a) Average Quantisation Error

(b) Topographic Error

Figure 4.1: Plots for quantisation and topographic errors-

time to compute the result with the batch algorithm was 35 seconds, which was better than the sequential algorithm that took about 203 seconds. Figures 4.1(a) and 4.1(b) present plots for the QE and TE, respectively, using different sample sizes of the *corpus*. For both plots, the errors were calculated for different map sizes: small (blue), normal (green) and big (red). As shown in the QE plot 4.1(a), the error decreases when the training is computed for big map and the number of elements used is either very large or corresponds to the entire input matrix. The TE 4.1(b) also improves when the map is trained with same configurations. So, we concluded that the best results were obtained for a big map and a large portion of the collection. Since the time to compute a large portion of the collection or the entire collection is very similar, further tests are trained with the entire input matrix.

Map Size	QE	TE	Field Purity	Empty Rate
Small	1.488	0.031	0.56	0.04
Normal	1.329	0.017	0.65	0.26
Big	1.076	0.022	0.79	0.44

Table 4.10: Abstracts results

Training duration	QE	TE	Field Purity	Empty Rate
Short	1.156	0.022	0.78	0.61
Normal	1.076	0.022	0.79	0.44
Long	1.072	0.008	0.79	0.44

Table 4.11: Training Duration.

Tables 4.10 and 4.11 illustrate the measured results and SOM errors for different map sizes and training duration. The Field Purity also enhances when the map is big, but as the size of the map increases the number of empty neurones also increase. Although the results improve when the map is big, training with an even bigger map size than those tested will produce a very large number of empty neurones with topological relations between related but distinct topics. In other words, the topological relation between clusters can be lost because each one will tend to be surrounded by empty neurones. An example of this emerging property can is shown at figure 4.2 from section 4.6, where *network infrastructure* documents are grouped into the same location and surrounded by empty neurones. The results performed better with the big map with a long training duration, therefore, computations of the document map are trained with these configurations.

4.5 Comparing RSOM with WEBSOM

For this thesis, a package was developed with the basic WEBSOM functions to produce the document map. The implemented package consists in the architecture studied in section 2.3, however, we only implemented the necessary functions to produce the document map and label the top words for each neurone.

	Quantisation Error	Topographic Error
WEBSOM	0.011	0.017
RSOM	1.072	0.008

Table 4.12: Quantisation and Topographic Errors for the Abstract collection.

A comparison between the output produced by the WEBSOM package and the RSOM approach was performed to determine the main differences between these systems. This analysis was based on results obtained with the Abstract *corpus*, where both maps were trained with the same configurations. Table 4.12 presents the training quantisation and topographic errors for both methods. The topographic error is similar between both, however, the map trained by WEBSOM had a much lower quantisation error than RSOM. This can be related to the dimension reduction accomplished on the WEBSOM which trained the map with 370 features, while RSOM trained the map with a dimension equal to the number of documents in the collection (1300 features).

	Field Purity	Empty Rate
WEBSOM	0.73	0.25
RSOM	0.79	0.44

Table 4.13: Field Purity and Empty Rate for the Abstract collection.

Table 4.13 illustrates the Field Purity and Empty Rate for both approaches. The map trained by WEBSOM has less empty neurones, however, the labeled neurones accuracy improves with the RSOM method. Since WEBSOM is based on single words, the semantic sharpness to describe documents is less accurate than RSOM, because this one uses multi-words to characterise documents. For example, the word "theorems" is relevant word but the Abstract collection can become ambiguous because there are several types of theorems that belong to different research fields. Thus, this can be a reason that different research areas are related. On the other hand, since RSOM resolves this issue because the REs are more accurate, specific terms usually do not collide within research areas.

For instance, the RE "limit theorems" is a variant of "theorems" which had the ambiguity solved and was related to just one research field.

	Precision	Recall
WEBSOM	0.71	0.52
RSOM	0.88	0.63

Table 4.14: Precision and Recall for the Abstract collection.

The neurones REs quality was measured through Precision and Recall values. The Precision (equation 4.3) estimates the REs quality associated to the neurone.

$$Precision = \frac{|\{\text{retrieved REs}\} \cap \{\text{good REs}\}|}{|\{\text{retrieved REs}\}|} \tag{4.3}$$

For example, if two of the n retrieved REs for a neurone, are incorrect REs, then Precision will be $(n-2)/n$. The Recall (equation 4.4) measures the ratio between the number of the best REs which are in the group of the n retrieved REs, and n.

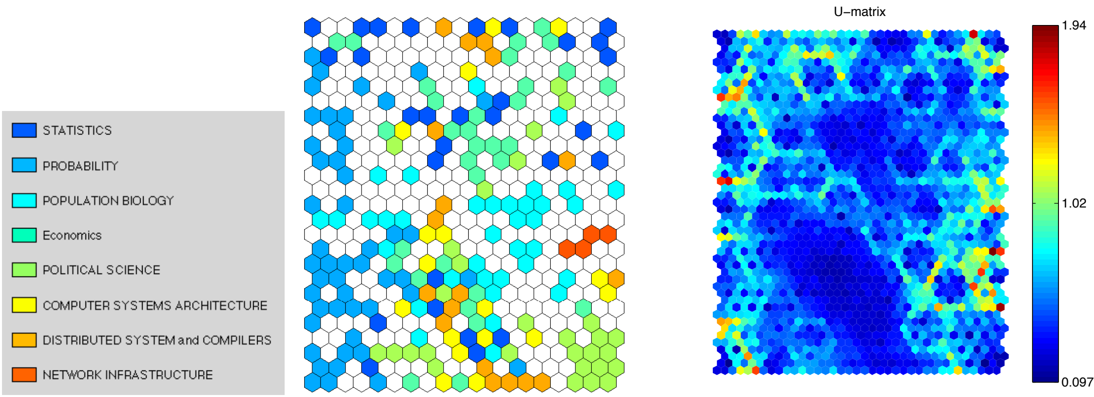
$$Recall = \frac{|\{\text{retrieved REs}\} \cap \{\text{the best REs}\}|}{|\{\text{retrieved REs}\}|} \tag{4.4}$$

For example, if the approach retrieves n REs to describe the neurone, but only two of them belong to the best n REs to describe the neurone, then Recall equals $2/n$.

Table 4.14 presents the Precision and Recall values for both approaches. The Precision suggests that both approaches organise correctly documents with good features. However, the WEBSOM had some examples that were vague and hold poor meaning. The RSOM performed better in this sense because the REs were more descriptive. The Recall value was also better for RSOM. This can be related due to the fact that WEBSOM uses single word to label neurones, which becomes much easier to find better words in the document to describe it. Besides, there were many suggested single words that could not be considered as the best ones to describe the neurones because individually they were vague, but if combined with another word they could have been consider as the best descriptor. This did not occur with REs that either were just a good RE or one of the best RE to describe the neurone. So, this results suggests it is better to use multi-words to describe the neurones. Nevertheless, the Recall values demonstrate that both systems could have a better selection for the best expressions to describe neurones.

4.6 Document Maps

Figures 4.2, 4.3 and 4.4 present the document maps for each collection studied in this thesis. Each figure presents a field map and respective U-Matrix (section 2.1.5). The field map is a projection of the network were each neurone is labeled according to their field. The U-Matrix provides a distance based map to visualise the distance between neurones of the network. Tables 4.15, 4.16 and 4.17 present samples of the neurones content and respective labeled neurone REs.



(a) Field Map

(b) U-matrix

Figure 4.2: Document map for the abstracts collection.

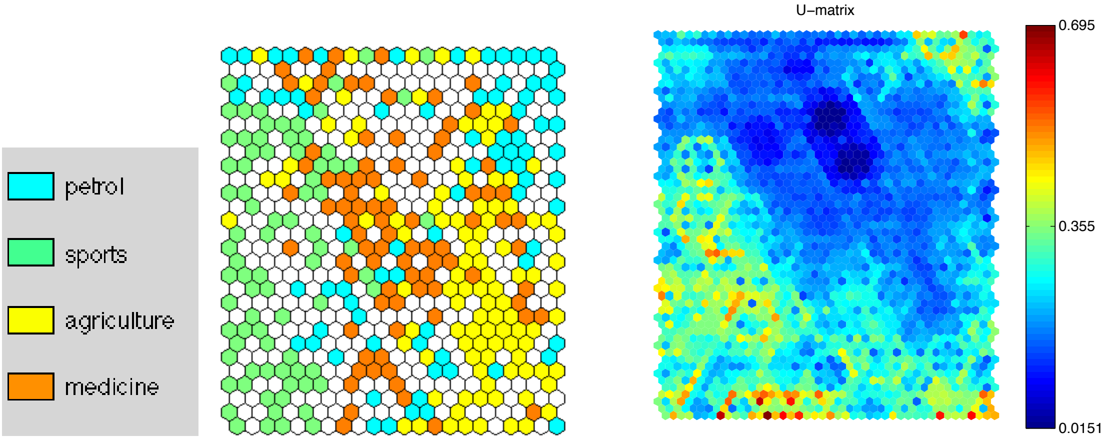
The Abstract collection produced a document map where documents were grouped by similar topics into the same location of the network. Table 4.15 illustrates some samples where similar documents are grouped into the same area of the map. In addition, the map also organised some locations by the same *research field*. For instance, *network infrastructure* documents occur only in one area of the map and form their own *cluster*. On the other hand, some clusters were composed by neurones of different research fields. This occurs because some research fields should in fact intersect, due to REs in common. An example of this property had *computer systems architecture*, and *distributed system and compilers* documents grouped by the RE "workstation clusters" into the same location. After an analysis of these documents, we concluded that they were in fact correlated and that "workstation clusters" was the best RE to define this neurone. However, in some cases neurones were labeled by good REs but when their description was analysed, they were poorly related; see example in the last row of table 4.15, where the name of one of the documents is "Comparative Political Economy" and one of the REs is "developing countries". So, in some cases the system can group documents incorrectly even if they have good REs in common.

Figure 4.3 presents the document map for the News collection. The field associated to a news is the domain which this one belongs. The different types of domains in this collection are: *petrol*; *sports*; *agriculture*; and *medicine*. The Empty Rate and Field Purity for the News collection were 43% and 88%, respectively. The Empty Rate was similar to the one measured for the Abstract *corpus* and their distribution on the map had a similar behaviour, where similar topics tend to form a cluster surrounded by empty neurones. The Field Purity was better than the one obtained for the Abstract collection because the fields for the News collections had small or no relation from one to another. So, it made sense that the News collection had a better Field Purity because similar documents should belong to the same field. This result matched with the goal meant for this collection, that was to produce an organised document map for random news extracted from an online

Neurone	
Documents	**Relevant Expressions**
P: Analysis and Geometry of Markov Chains Diffusion Processes P: Lower Tail Probabilities and Limit Theorems in Probability and Statistics P: Mathematical Sciences: Gaussian Measures and Small Ball Probabilities P: Mathematical Sciences: Some Problems in Probability Theory P: Fluctuations in Asymmetric Processes with Static and Dynamic Random Environments P: Limits and Deviations for Interacting Random Systems P: Self-Normalized Limit Theorems and Small Ball Probabilities	*limit theorems* *gaussian processes* *fundamental importance* *probability and statistics*
PB: Genetic Diversity in Different Forms of Rarity: Computer Models and Empirical Data from Lomatium PB: Collaborative Research: Sex Ratio Evolution in Ephemeral Demes of a Gynodioecious Plant PB: The Impact of Forest Fire Management on the Population Structure of Glade Species in the Ozarks	*conservation biology* *extinction and recolonization*
E: Emperical Analyses of Competitive Bidding E: Empirical Analyses of Competitive Bidding E: Empirical Analysis of Auctions E: Economic Analysis of Data from Laboratory Experiments: Endogeniety, Attrition, Subject Heterogeniety and Interdependencies	*winners curse* *ex ante*
E: Comparative Political Economy E: Collaborative Research: Nutritional Investments in Children Adult Human Capital and Adult Productivities PS: Environmental Regulation in Latin America: Economic Internationalization and Political Institutions	*developing countries* *interest groups*

Table 4.15: Samples of Abstract neurones

journal.



(a) Field Map

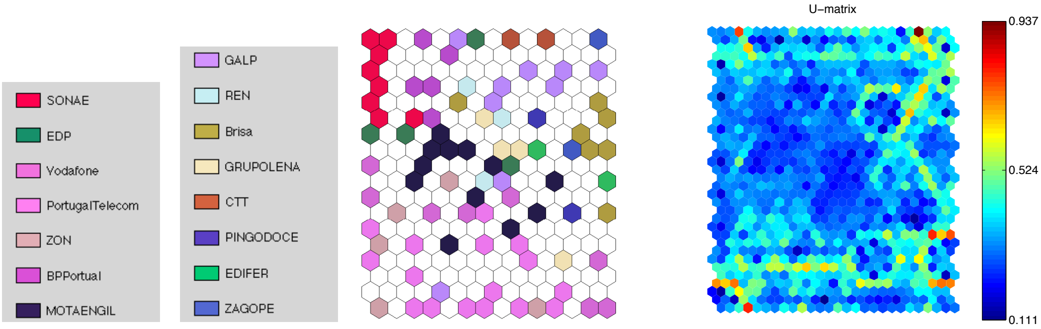
(b) U-matrix

Figure 4.3: Document map for the News collection.

Table 4.16 illustrates some results of the News document map. Although there were good results on this map, there were also some unexpected results related to news which could not be related to any other. In other words, there were some neurones that grouped documents because their attribute values were zero or very low and had no resemblance between them. The last row in Table 4.16 is an example where documents were grouped because of this absence.

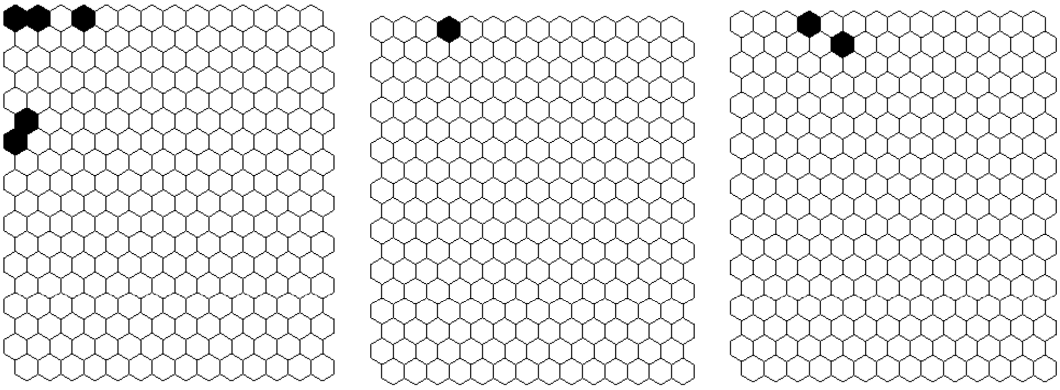
The document map produced by the Company Reports collection (figure 4.4) also organised similar topics into the same area of the map. In this collection, the field used to associate to a report was the company that released it. Thus, the number of fields was 15 companies and each one had around 10 reports associated to it. The Field Purity was 82% and the Empty Rate was 32%. The document distribution on the network was similar to the Abstract, where documents of the same field tend to be together. In addition, there were also some cases that were correctly intersected different fields in the same area. For example, the last row in table 4.17, intersected reports of two telecommunication companies.

The utility of Keyword component planes was already analysed in sec 3.3.2. Nevertheless, we wish to make a final validation of this useful tool in RSOM in Company Reports collections. Figure 4.5 illustrates three top keyword component places for the clusters BP and SONAE in the Company Reports collection. These REs exemplify a topographic relation between nearby areas of the map. Indeed "Golfo do México" and "Texas City" are highly relevant in the semantic context of BP oil activities: BP has several oil platforms in Gulf of Mexico and had problems in the Texas City refinery. At the same time, both SONAE and BP operate in the United Kingdom (Reino Unido).



(a) Field Map (b) Companies Reports

Figure 4.4: Document map for the Company Reports collection.



(a) Reino Unido (b) Texas City (c) Golfo do México

Figure 4.5: Selected Keyword Component Planes for the Company Reports Collection.

Neurone	
Documents	**Relevant Expressions**
Petrol: Guiné Equatorial, apreciada pelo petróleo e criticada pela ditadura Petrol: «Não vamos decidir nada sobre adesão» da Guiné Equatorial, diz Sócrates Petrol: Possível adesão da Guiné-Equatorial alimenta polémicas	*Guiné Equatorial* *Teodoro Obiang* *adesão da Guiné* *comunidade dos países de língua* *língua portuguesa cplp*
Sports: João Sousa e Maria João Koehler são os melhores tenistas portugueses Sports: Sharapova chega com recorde aos "quartos" do Open da Austrália Sports: Roger Federer alcança 250.ª vitória em torneios do Grand Slam Sports: Duas tenistas portuguesas no quadro principal do Open da Austrália	*Grand Slam* *Andy Murray* *Open da Austrália* *Ekaterina Makarova* *Roland Garros* *Novak Djokovic*
Agriculture: Governo da Madeira anuncia apoio para agricultores lesados Agriculture: Ministério quer apurar prejuízos em culturas afectadas pelo mau tempo Agriculture: Produtores preocupados com efeitos da chuva nas plantações Agriculture: Alimentos: «Produção nacional ajuda a travar aumento dos preços»	*gonçalo escudeiro* *seguros de colheita* *prazos dos seguros* *produtores hortofrutícolas*
Medicine: Bastonário da Ordem dos Médicos contesta curso de medicina em Aveiro Medicine: Mariano Gago diz que não há médicos no desemprego em parte nenhuma do mundo Medicine: Novo curso nasce hoje na Universidade de Aveiro Medicine: Primeiro curso de Medicina só para licenciados arrancou em Setembro no Algarve Medicine: Governo autoriza curso de Medicina na Universidade de Aveiro	*mariano gago* *ensino superior* *universidade de aveiro* *curso de medicina* *colaboração com a universidade do porto será formalizado* *ministro do ensino superior*
Agriculture: Crise limita adaptação de Portugal às alterações climáticas Medicine: Ordem dos Médicos pede investigação às receitas denunciadas pela ANF Medicine: Descobertos anticorpos capazes de proteger ratos contra doses mortais de vírus da gripe Sports: Euro2020 vai disputar-se em 13 cidades europeias	

Table 4.16: Samples of News neurones.

Neurone	
Documents	**Relevant Expressions**
BPPortugal 2005 BPPortugal 2006 BPPortugal 2007 BPPortugal 2008	*texas city* *golfo do méxico* *teor de carbono* *questão das alterações climáticas* *taxa de substituição*
Vodafone 1996 Vodafone 1997 Vodafone 1999 Vodafone 2002 Vodafone 2004 Vodafone 2010	*equipamentos terminais* *capitalização bolsista* *taxa de penetração* *telecel continuou a afirmarse* *valorização da telecel* *população total*
Galp 2007 Galp 2009 Zagobe 2007 Zagobe 2008 Zagobe 2010	*guiné equatorial* *biodiesel hidrogenado* *república do congo* *metro de lisboa* *determinação e firmeza* *bloco bms11*
PT 1999 ZON 2000 ZON 2001	*comércio electrónico* *prestado e confiança demonstrada* *elevadas taxas* *incentivo permanente* *velocidade à internet via*

Table 4.17: Samples of Company Reports neurones.

5

Conclusion

The approach proposed on this thesis has shown to be successful to produce a document map through documents characterised by relevant expressions. Based on experiments described on this dissertation and related research, it can be concluded that REs are semantically more accurate than single words to summarise the main topics described on textual documents. In addition, it was also seen that the document map produced with REs was more precise than the map produced by single words. Nevertheless, the WEBSOM publications and the results computed by the WEBSOM package developed in this thesis demonstrate that single words are good features to connect documents. Thus, this motivates for future work on the RSOM method to include unigrams.

LocalMaxs algorithm provided many expressions to describe documents for each collection. However, this extractor was not perfect and also considered incorrect expressions. In order to filter incorrect or weak expressions, on the present dissertation experiments were made in order to develop a metric to weight the relevance of terms in the collection. This metric seemed to provide good weighting of most informative expressions to characterise documents. However, it over penalises some correct MWEs that are not top informative. So, in addition to include unigrams in future work, it is also suggested to improve the MWEs filter of the RSOM approach.

The RSOM package had limitations due to the high dimensionality of features. Although this system performs a reduction of the vocabulary considering all expressions in the new set of features, their size is equal to the number of documents in the collection. So, this system is limited for a very high number of documents because the input structure is a document-by-document matrix. There are several types of dimension reduction that can be done over the matrix, such as principal component analysis or random mapping. However, since the main goal of this dissertation was to produce a document map based on REs, further dimension reductions to enable the system to support larger collections were considered for future work.

Self-organising maps were an effective algorithm to organise the data into neural network with topological relation between them. Results could be improved by working in the context of SOM in what concerns its internal method to evaluate distances. Moreover, since the attributes on the RSOM method stand for correlations between documents, it should make sense to consider the high correlations more influent for the best matching than low values, as these are just absence of similarity.

As shown on section 4.6, the top keyword component planes revelled themselves as a valuable tool for helping to describe, localise and contextualise the documents in the RSOM map. Results have also shown that this contribution is language and topic domain independent.

Although this dissertation has chosen an academic pathway, acquired results on business test bed documents show the exceptional suitability of RSOM for building the initial and visionary Corporate Semantic Map as it was first envisioned by InspirennovIT. Further work is now under way for incorporating acquired Corporate Semantic Maps inside the Best Supplier project.

Bibliography

[1] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, “Creating an order in digital libraries with self-organizing maps,” in *Proceedings of WCNN'96, World Congress on Neural Networks, September 15-18, San Diego, California*, pp. 814–817, Mahwah, NJ: Lawrence Erlbaum and INNS Press, 1996.

[2] K. Lagus, S. Kaski, and T. Kohonen, “Mining massive document collections by the websom method,” *Inf. Sci.*, vol. 163, pp. 135–156, 2004.

[3] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, V. Paatero, and A. Saarela, “Self organization of a massive document collection,” *IEEE Transactions on Neural Networks*, vol. 11, pp. 574–585, 2000.

[4] S. Alves, P. Bengala, J. Petrucci, and N. Marques, “Applying websom for building a corporate semantic map,” 2012.

[5] E. Alpaydin, *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, second ed., 2004.

[6] R. Sibson, “Slink: An optimally efficient algorithm for the single-link cluster method,” *The Computer Journal*, vol. 16, pp. 30–34, 1973.

[7] D. Defays, “An efficient algorithm for a complete link method,” *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 1977.

[8] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press, 1967.

[9] F. Aurenhammer, “Voronoi diagrams – a survey of a fundamental geometric data structure,” *ACM COMPUTING SURVEYS*, vol. 23, no. 3, pp. 345–405, 1991.

[10] “UCI Machine Learning Repository: Iris Data Set,” 2007.

[11] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2005.

[12] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The Computer Journal*, vol. 41, pp. 578–588, 1998.

[13] T. Kohonen, M. R. Schroeder, and T. S. Huang, eds., *Self-Organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 3rd ed., 2001.

[14] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-organizing map in matlab: the som toolbox," in *In Proceedings of the Matlab DSP Conference*, pp. 35–40, 2000.

[15] Y. Cheng, "Convergence and ordering of kohonen's batch map," *Neural Comput.*, vol. 9, no. 8, pp. 1667–1676, 1997.

[16] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Som toolbox for matlab 5," 2000.

[17] A. Ultsch and H. P. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," in *Proceedings of International Neural Networks Conference (INNC)*, (Paris), pp. 305–308, Kluwer Academic Press, 1990.

[18] J. Vesanto, "Som-based data visualization methods," *Intelligent Data Analysis*, vol. 3, pp. 111–126, 1999.

[19] A. Ultsch, "U*-matrix: a tool to visualize clusters in high dimensional data.," *Computer*, no. 36, pp. 1–12, 2003.

[20] A. Ultsch, "Pareto density estimation: A density estimation for knowledge discovery," in *Innovations in Classification, Data Science, and Information Systems - Proceedings 27th Annual Conference of the German Classification Siciety (GfKL'03)*, pp. 91–100, 2003.

[21] T. Honkela, "Self-organizing maps of words for natural language processing applications," in *In Proceedings International ICSC Symposium on Soft Computing*, 1997.

[22] M. Pöllä, T. Honkela, and T. Kohonen, "Bibliography of self-organizing map (som) papers: 2002–2005 addendum."

[23] K. Lagus, *Text Mining with the WEBSOM*. Acta polytechnica scandinavica ma 110, Neural Networks Research Centre, Helsinki University of Technology, Finland, 2000.

[24] P. Andritsos, P. Tsaparas, R. J. Miller, and K. C. Sevcik, "scalable clustering of categorical data," in *In EDBT*, pp. 123–146, 2004.

[25] D. Gibson, J. Kleinberg, and P. Raghavan, “Clustering categorical data: An approach based on dynamical systems,” pp. 311–322, 1998.

[26] S. Guha, R. Rastogi, and K. Shim, “Rock: A robust clustering algorithm for categorical attributes,” in *In Proc.ofthe15thInt.Conf.onDataEngineering*, 2000.

[27] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[28] S. Guha, R. Rastogi, and K. Shim, “Cure: An efficient clustering algorithm for large databases.,” in *SIGMOD Conference* (L. M. Haas and A. Tiwary, eds.), pp. 73–84, ACM Press, 1998.

[29] R. T. Ng and J. Han, “Efficient and effective clustering methods for spatial data mining,” in *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB ’94, (San Francisco, CA, USA), pp. 144–155, Morgan Kaufmann Publishers Inc., 1994.

[30] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: an efficient data clustering method for very large databases,” in *SIGMOD*, pp. 103–114, 1996.

[31] J. F. d. Silva, J. a. Mexia, C. A. Coelho, and J. G. P. Lopes, “Document clustering and cluster topic extraction in multilingual corpora,” in *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM ’01, (Washington, DC, USA), pp. 513–520, IEEE Computer Society, 2001.

[32] G. Salton, A. Wong, and C. Yang, “A vector space model for automatic indexing,” *Commun. ACM*, vol. 18, pp. 613–620, 1975.

[33] M. Dash and H. Liu, “Feature selection for clustering,” pp. 110–121, Springer-Verlag, 2000.

[34] I. T. Jolliffe, *Principal Component Analysis*. Springer, second ed., 2002.

[35] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, vol. 41, no. 6, pp. 391–407, 1990.

[36] S. Kaski, “Dimensionality reduction by random mapping: Fast similarity computation for clustering,” 1998.

[37] K. Lagus, “Map of WSOM’97 abstracts—alternative index,” in *Proceedings of WSOM’97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pp. 368–372, Espoo, Finland: Helsinki University of Technology, Neural Networks Research Centre, 1997.

[38] S. Kaski, “Data exploration using self-organizing maps,” *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, March 1997. DTech Thesis, Helsinki University of Technology, Finland.

[39] H. Fang, T. Tao, and et al., "A formal study of information retrieval heuristics," 2004.

[40] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, (New York, NY, USA), pp. 232–241, Springer-Verlag New York, Inc., 1994.

[41] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Computer Journal*, vol. 26, no. 4, pp. 354–359, 1983.

[42] P. Willett, "Recent trends in hierarchic document clustering: a critical review," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 577–597, 1988.

[43] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in *Data Mining and Knowledge Discovery*, pp. 515–524, ACM Press, 2002.

[44] C. van Rijsbergen and W. Croft, "Document Clustering: An Evaluation of Some Experiments with the Cranfield 1400 Collection," *Information Processing and Management*, vol. 11, pp. 71–182, 1975.

[45] N. Jardine and C. J. van Rijsbergen, "The use of hierarchical clustering in information retrieval," *Information Storage and Retrieval*, vol. 7, pp. 217–240, 1971.

[46] J. H. Gennari, P. Langley, and D. Fisher, "Models of incremental concept formation," *Artificial Intelligence*, vol. 40, no. 1-3, pp. 11–61, 1989.

[47] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," in *Machine Learning*, pp. 139–172, 1987.

[48] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," in *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, (New York, NY, USA), pp. 357–366, ACM, 2006.

[49] C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 2, pp. 245–255, 2004.

[50] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/gather: a cluster-based approach to browsing large document collections," in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, (New York, NY, USA), pp. 318–329, ACM, 1992.

[51] O. Zamir and O. Etzioni, "Web document clustering: a feasibility demonstration," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and*

development in information retrieval, SIGIR '98, (New York, NY, USA), pp. 46–54, ACM, 1998.

[52] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "Newsgroup exploration with WEBSOM method and browsing interface," Tech. Rep. A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996.

[53] T. Kohonen, S. Kaski, K. Lagus, and T. Honkela, "Very large two-level SOM for the browsing of newsgroups," in *Proceedings of ICANN96, International Conference on Artificial Neural Networks, Bochum, Germany, July 16-19, 1996* (C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, eds.), Lecture Notes in Computer Science, vol. 1112, pp. 269–274, Berlin: Springer, 1996.

[54] H. Ritter and T. Kohonen, "Self-organizing semantic maps," *Biological Cybernetics*, vol. 61, pp. 241–254, 1989.

[55] T. Honkela and P. V. T. Kohonen, "Contextual relations of words in Grimm tales, analyzed by self-organizing map," in *Proc. ICANN'95, International Conference on Artificial Neural Networks* (F. Fogelman-Soulié and P. Gallinari, eds.), vol. II, pp. 3–7, EC2, 1995.

[56] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "WEBSOM—self-organizing maps of document collections," in *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pp. 310–315, Espoo, Finland: Helsinki University of Technology, Neural Networks Research Centre, 1997.

[57] T. Kohonen, "Self-organization of very large document collections: State of the art," in *Proceedings of ICANN98, the 8th International Conference on Artificial Neural Networks* (L. Niklasson, M. Bodén, and T. Ziemke, eds.), vol. 1, pp. 65–74, London: Springer, 1998.

[58] J. J. Väyrynen and T. Honkela, "Word category maps based on emergent features created by ICA," in *Proceedings of the STeP'2004 Cognition + Cybernetics Symposium* (H. Hyötyniemi, P. Ala-Siuru, and J. Seppänen, eds.), Publications of the Finnish Artificial Intelligence Society, pp. 173–185, Finnish Artificial Intelligence Society, 2004.

[59] G. Salton and M. J. Mcgill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.

[60] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "Exploration of full-text databases with self-organizing maps," in *Proceedings of the ICNN96, International Conference on Neural Networks*, vol. I, pp. 56–61, Piscataway, NJ: IEEE Service Center, 1996.

[61] "Websom - self-organizing maps for internet exploration."

[62] *Keyword selection method for characterizing text document maps*, vol. 1, 1999.

[63] M. A. Finlayson and N. Kulkarni, “Detecting multi-word expressions improves word sense disambiguation,” in *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, (Stroudsburg, PA, USA), pp. 20–24, Association for Computational Linguistics, 2011.

[64] O. C. Acosta, A. Villavicencio, and V. P. Moreira, “Identification and treatment of multiword expressions applied to information retrieval,” in *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, MWE ’11, (Stroudsburg, PA, USA), pp. 101–109, Association for Computational Linguistics, 2011.

[65] B. Daille, “Study and implementation of combined techniques for automatic extraction of terminology,” in *The Balancing Act: Combining Symbolic and Statistical Approaches to Language* (J. Klavans and P. Resnik, eds.), Cambridge, MA: MIT Press, 1996.

[66] A. Copestake, F. Lambeau, F. B. Aline Villavicencio, T. Baldwin, I. A. Sag, and D. Flickinger, “multi-word expressions: linguistic precision and reusability,” in *Proceedings of the Third conference on Language Resources and Evaluation*, pp. 1941–1947, 2002.

[67] G. Dias, “Multiword unit hybrid extraction,” in *Workshop on Multiword Expressions of the 41st ACL meeting*, pp. 41–48, 2003.

[68] N. Kulkarni and M. A. Finlayson, “jmwe: A java toolkit for detecting multi-word expressions,” in *Proceedings of the Workshop on multi-word Expressions: from Parsing and Generation to the Real World. Association for Computational Linguistics*, pp. 122–124, 2011.

[69] R. Mahesh and K. Sinha, “Stepwise mining of multi-word expressions in hindi,” in *Proceedings of the Workshop on multi-word Expressions: from Parsing and Generation to the Real World. Association for Computational Linguistics*, pp. 110–115, 2011.

[70] S. Martens and V. Vandeghinste, “An efficient, generic approach to extracting multiword expressions from dependency trees,” in *Proceedings of the Workshop on multiword Expressions: from Theory to Applications*, pp. 84–87, 2010.

[71] E. Wehrli, V. Seretan, and L. Nerima, “Sentence analysis and collocation identification,” in *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, (Beijing, China), pp. 27–35, Association for Computational Linguistics, August 2010.

[72] K. W. Church and P. Hanks, “Word association norms, mutual information, and lexicography,” *Computational Linguistics*, vol. 16, pp. 22–29, Mar. 1990.

[73] W. A. Gale and K. W. Church, "Concordance for parallel texts," in *Proceedings of the Seventh Annual Conference of the UW Centre of the new OED and Text Research, Using Corpora*, EPIA '99, pp. 40–62, 1991.

[74] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational Linguistics*, vol. 19, pp. 61–74, Mar. 1993.

[75] J. F. Silva and G. P. Lopes, "A local maxima method and a fair dispersion normalization for extracting multiword units," in *Proceedings of the 6th Meeting on the Mathematics of Language*, pp. 369–381, 1999.

[76] J. F. Silva, G. Dias, S. Guillor, and G. P. Lopes, "Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units," in *Proceedings of the 9th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*, EPIA '99, (London, UK), pp. 113–132, Springer-Verlag, 1999.