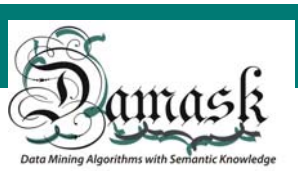

TIN2009-11005
DAMASK

Data-Mining Algorithms with Semantic Knowledge

PROYECTO DE INVESTIGACIÓN
PROGRAMA NACIONAL DE INVESTIGACIÓN FUNDAMENTAL,
PLAN NACIONAL DE I+D+i 2008-2011
ÁREA TEMÁTICA DE GESTIÓN: Tecnologías informáticas

Deliverable D3

State of the art of clustering algorithms and semantic similarity measures

Authored by
Montserrat Batet, Universitat Rovira i Virgili
David Sánchez, Universtitat Rovira i Virgili
Aïda Valls, Universitat Rovira i Virgili

Document information

project name:	DAMASK	
Project reference:	TIN2009-11005	
type of document:	Deliverable	
file name:	D3.pdf	
version:	Final	
authored by:	M. Batet, D. Sánchez, A. Valls	12/11/2010
co-authored by		
released by:	D. Sánchez	09.12.2010
approved by:	Co-ordinator	Antonio Moreno

Document history

version	date	reason of modification
1.0	06.July.2010	Classification of clustering algorithms. Clustering software tools. State of the art of semantic similarity measures.
2.0	04.Oct.2010	First review and new materials to complete some sections.
2.1	07.Nov.2010	Format and references checking
3.0	12.Nov.2010	Improvements in discussion sections. Inclusion of the third component: link between semantic similarity and clustering.
4.0	09.Dec.2010	Final revision and formatting

Table of Contents

1	**Introduction**	**3**
2	**Survey of clustering algorithms**	**4**
2.1	Notation	5
2.2	Types of Unsupervised Clustering Algorithms	5
2.2.1	Partitional clustering	6
2.2.2	Hierarchical clustering	8
2.2.2.1	Agglomerative clustering	8
2.2.2.2	Divisive clustering	12
2.2.3	Other Clustering techniques	12
2.3	Discussion	13
3	**Software tools for clustering**	**15**
4	**Semantic similarity measures**	**19**
4.1	Ontology-based measures	21
4.1.1	Edge counting-based measures	21
4.1.2	Feature-based measures	24
4.1.3	Information Content-based measures	26
4.2	Distributional approaches	28
4.2.1	First order co-occurrence	29
4.2.2	Second order co-occurrence	32
4.3	Evaluation	35
4.4	Discussion	37
5	**Semantic similarity measures into clustering algorithms**	**40**
6	**References**	**41**

1 Introduction

This document is the first deliverable of the Task 2 of the DAMASK project. Task T2 is focused on the goal O2 of the project: design of a clustering method based on ontologies. The inputs of this task will be (1) a data matrix object × attribute (e.g. touristic destinations) and (2) a domain ontology. Based on those inputs, a method will be designed for automatically building clusters with the help of the contextual knowledge provided by the domain ontology. Moreover, an automatic interpretation process of the clusters will also be studied, in order to obtain a semantic description of the clusters that can help the user in his/her decision making tasks.

This deliverable is the output of the subtask (T2-1): *State of the art about the techniques for automatic clustering of data and about the existing methods for similarity measurement for semantic concepts*. The complete schedule of the tasks is given in Figure 1.

This document is divided into three main parts:

1. A study of the traditional clustering methods, which do not use contextual semantic knowledge to guide the process of classification. Advantages and drawbacks have been reported.
2. A study of the similarity measures that are being used for comparing a pair of concepts from a domain-specific ontology or general-purpose semantic structures like WordNet.
3. Evaluation of the applicability of those semantic similarity measures into the clustering algorithms.

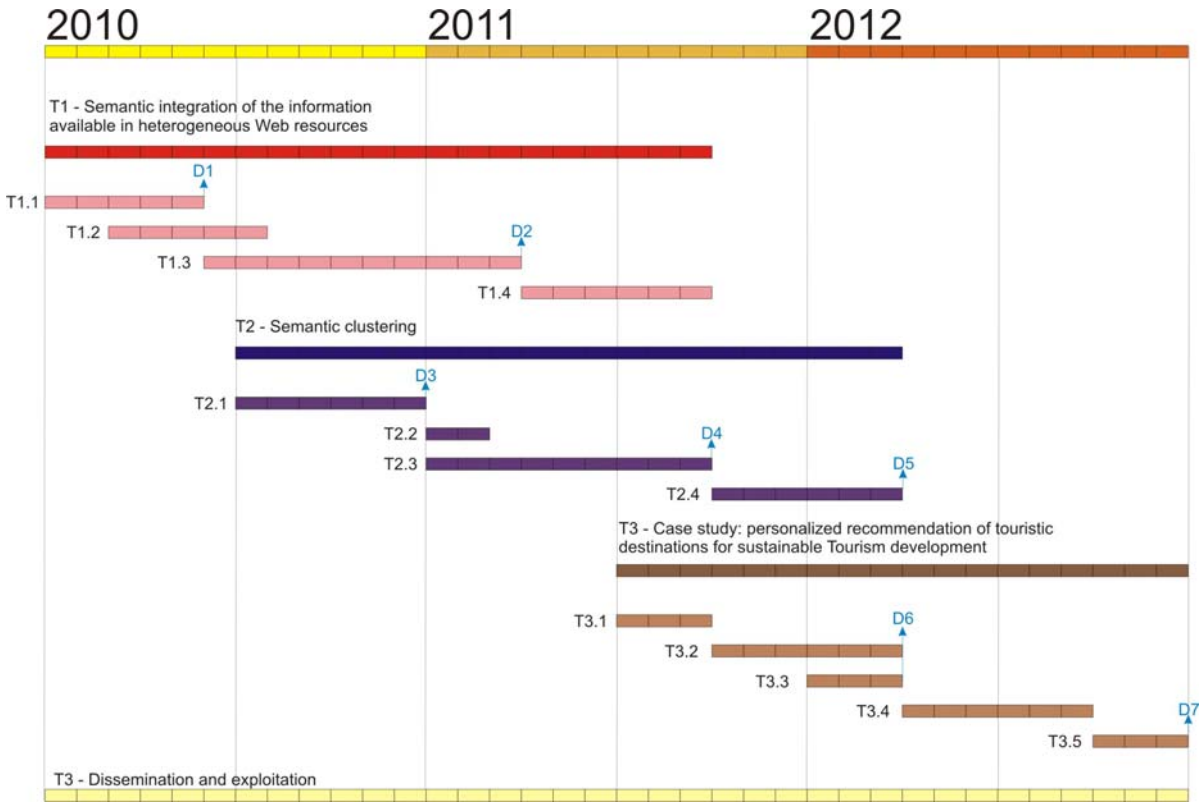

Figure 1: Tasks of DAMASK

2 Survey of clustering algorithms

The goal of this section is to provide a review of clustering techniques. Clustering is generally seen as the task of finding groups of similar individuals based on information found in data, which means that the data individuals in the same group are more similar to each other than to individuals in other groups. So, clustering algorithms partition data into a certain number of clusters (groups, subsets, or categories) (Xu and Wunsch, 2005). Clustering algorithms try to minimize the dispersion inside each group. Thus, the goal is to build clusters with a great similarity (homogeneity) within the members that form the group and a great distance between members of different groups.

Clustering methods are unsupervised techniques that aim to discover the structure of a data set. This approach must be distinguished from *Classification* or *supervised methods*, which learn how to assign instances to predefined classes or categories. In the latter model, the classifier is trained using data from the different classes. So, a (training or learning) set of labeled objects is used to build a classifier for the categorization of future observations. A third typology is denoted as *semi-supervised clustering*. These algorithms try to improve the results of the unsupervised methods adding some extra knowledge of the experts. Those methods explore different approaches to guide the clustering process, like the introduction of different types of constraints (Basu et al., 2008; H. Huang et al., 2008) (e.g. cluster size balancing, pairwise constraints for object's relationships) or the use of domain-dependent rules (Gibert et al., 2010; Valls et al., 2009). It has been seen that the use of this additional background knowledge helps to improve the coherence of the obtained results.

For the purposes of this project, we will work with unsupervised clustering methods. So, the rest of this document is focused on this type of methods.

Clustering is a masterpiece in many data mining methodologies, because it helps to discover new knowledge from unstructured data sets. A definition of data mining and knowledge discovery, made in (Fayyad et al., 1996), is: "*The non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data".*

In fact, clustering has been used in many data mining problems, such as to build a structure of a complex data set, to reveal associations between objects or to make generalizations. Some exemplary problems illustrating the role of clustering in these tasks is given in (Mirkin, 2005). Clustering methods have been practically applied in a wide variety of fields, ranging from engineering (*e.g.* pattern recognition, mechanical engineering, electrical engineering), computer sciences (*e.g.* web mining, spatial database analysis, image segmentation, privacy), life and medical sciences (*e.g.* genetics, biology, microbiology, paleontology, psychiatry, pathology), to earth sciences (*e.g.* geography. geology, remote sensing), social sciences (*e.g.* sociology, psychology, archeology, education), and economics (*e.g.* marketing, business)). Recently, new fields of application have increased the research on this topic, specially due to the developments in information retrieval and text mining, spatial database applications (Fan, 2009; Han et al., 2001; Monreale et al., 2010), Web applications (Cadez et al., 2003; Carpineto et al., 2009; Kimura et al., 2010) and DNA analysis in computational biology (J. Y. Chen and Lonardi, 2009; Romdhane et al., 2010; Tirozzi et al., 2007), among others.

2.1 Notation

Individuals: are the objects that are being evaluated and grouped in the clustering. They can also be called instances, cases, patterns, tuples, …. Individuals will be referenced as $I = \{1,...,N\}$.

Features: are the properties that describe the individuals. Each feature is treated as an independent variable (i.e. attribute or dimension) in the space of representation of the individuals. We will consider K features, denoted as $X_1, ..., X_k$. Different types of features can be distinguished. They have been characterized in the document (Batet et al., 2010).

Data Matrix: is a matrix that contains the values of all the features for a set of N objects. This matrix has a dimension of (N,K) and is defined as follows:

$$X = \begin{pmatrix} x_{11} & x_{12} & ... & x_{1K} \\ x_{21} & x_{22} & ... & x_{2K} \\ ... & ... & ... & ... \\ x_{N1} & x_{N2} & ... & x_{NK} \end{pmatrix}$$

Notice that each row corresponds to an individual, represented as a multidimensional vector $X = (x_{i1}, x_{i2}, ..., x_{iN})$ where x_{ik} is the value of the feature k taken by object i.

2.2 Types of Unsupervised Clustering Algorithms

Clustering algorithms can be divided in function of the properties of the generated clusters into hierarchical clustering and partitional clustering.

- **Partitional clustering** is the division of the set of data objects into non-overlapping subsets such that each data object is exactly in one subset. So, it attempts to find a C-partition of I where C is a pre-specified number indicating the amount of desired clusters ($C \leq N$).
- **Hierarchical clustering** attempts to construct a tree-like nested structure partition of I. These methods create a hierarchical decomposition of the given data set, producing a binary tree known as a *dendogram*. The *root* node of the dendrogram represents the whole data set X and each *leaf* node is a single object i; the rest of intermediate nodes correspond to clusters that group similar objects. The tree is a taxonomy with is-a relations. Overlapping between clusters is not admitted. The clusters in the dendogram can have an associated numerical value. This value indicates the degree of proximity between the objects, which is related with the intra-cluster cohesion.

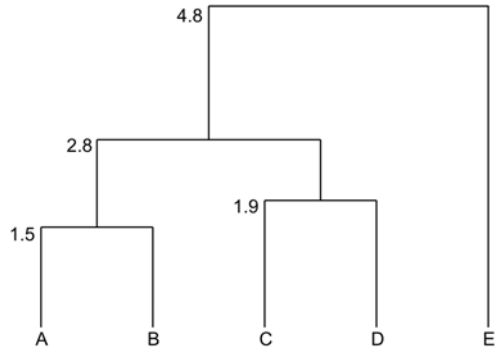

Figure 1: Dendogram.

2.2.1 Partitional clustering

Partitional clustering assigns a set of *N* objects into *C* clusters with no hierarchical structure where each group must contain at least one object and each object must belong to one group. It is important to note that in this clustering approach, the number of clusters is defined in advance. It is usually done on the basis of some specific criterion, so one of the important factors in partitional clustering is the *criterion* function (Hansen and Jaumard, 1997).

Partitioning methods are divided into two major subcategories:

- The *centroid* algorithms represent each cluster by using the centre of gravity of the objects, with a artificially created prototype. This approach has the problem of defining a method for generating this prototype, which is usually based on calculating some sort of average of the values of the objects. The definition of an averaging function hampers the application to non-numerical variables. Different approaches have been defined using dissimilarity measures for categorical variables, such as Huang (Z. Huang, 1998) and Gupta *et al.* (Gupata et al., 1999). If an ordinal relation can be defined on the categorical values, then specific averaging operators are defined (Godo and Torra, 2000). Other solution consists in using median operators to build the prototype. (Beliakov et al., 2010; Domingo-Ferrer and Torra, 2003).
- The *medoid* algorithms represent each cluster by means of the object of the cluster whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the cluster. This approach avoids the problem of calculating an artificial prototype. It only requires the definition of a distance between objects.

The most important algorithm for partitional clustering is called k-means. It is present in the major statistical software packages, as it will be seen in section 3. Several variations of this algorithm can be found. They are reviewed in this section.

- *k-means*: it is the most well-known centroid algorithm (Forgy, 1965; MacQueen, 1967). K-means attempts to find a number *k* of clusters fixed a priori, which are represented by its centroid. K-means uses the squared error criterion (MacQueen, 1967) as criterion function.

The steps of this clustering algorithm are the following:

A priori: Determine the number K of partitions.

Step 1) Generate a k-partition randomly or based on some prior knowledge and calculate the cluster prototype matrix (cluster centroids).

Step 2) Assign each object X_i to the nearest cluster (i.e. the cluster prototype or centroid).

Step 3) Recalculate the centroid of each cluster based on the current partition.

Step 4) Repeat steps 2)–3) until there is no change for each cluster or when a number of beforehand defined iterations is done.

The advantages of this algorithm are its simplicity and its time complexity (can be used to cluster large data sets). Its stopping criteria usually needs a small number of iterations making this algorithm very efficient. The disadvantages of this method are: (1) it is sensitive to the selection of the initial partition and there is no efficient method for identifying the initial partitions and the number of clusters. Usually, the strategy followed is to run the algorithm iteratively using different random initializations. However, some authors studied the initialization of the method (Kaufman and Rousseeuw, 1990; Mirkin, 2005). (2) The iterative procedure of k-means cannot guarantee convergence to a global optimum (minimum global variance, although it can guarantee the minimum variance inside of a cluster or local optimum). (3) Due to its initial randomness, obtaining the same results for all the executions cannot be guaranteed. (4) K-means is sensitive to outliers and noise because even if an object is quite far away from the cluster centroid, it is still forced to be in the cluster, which distorts the cluster shapes.

However, there are variants of the k-means which solve some of these limitations. In the following we briefly mention them:

- PAM (Kaufman and Rousseeuw, 1990) (partitioning around medoids) is an early k-medoid algorithm that uses the data points (medoids) as the cluster prototypes avoiding the effect of outliers. PAM has a drawback that it works inefficiently for a large data set due to its time complexity (Han et al., 2001).
- CLARA: (Kaufman and Rousseeuw, 1990) was developed to solve the problem of a large data set. CLARA applies the PAM to sampled objects instead of all objects.
- *ISODATA* algorithm (iterative self-organizing data analysis technique) (Ball and Hall, 1965): it is a variation of the k-means that employs a technique of merging and splitting clusters. A cluster is split when its variance is above a pre-specified threshold, and two clusters are merged when the distance between their centroids is below another pre-specified threshold. So, ISODATA can estimate the number of clusters with these merging and splitting procedures. ISODATA considers the effect of outliers in clustering procedures
- GKA (genetic -means algorithm)(Krishna and Murty, 1999): it is designed in order to avoid getting stuck in a local optimum, it can find a global optimum.
- The *k-modes algorithm* (Z. Huang, 1998): it uses the simple matching coefficient measure to deal with categorical attributes.

- The *k-prototypes algorithm* (Z. Huang, 1998): this algorithm through the definition of a combined dissimilarity measure, further integrates the k-means and k-modes algorithms to allow for clustering instances described by mixed attributes.

- The *X-means algorithm* (Pelleg and Moore, 2000): this method automatically finds the number of clusters by using a binary k-means, combined with internal validity indices. At each step a k-means with K = 2 is executed to find a division in two clusters. If the split increases the overall value given by the internal validity indices, the cluster is split and the binary k-means continues execution, recursively. If it is no possible to divide any cluster obtaining an improved validity index, the algorithm stops and takes the current partition as result.

2.2.2 Hierarchical clustering

Hierarchical clustering (HC) algorithms organize data into a hierarchical structure according to a proximity matrix. A proximity matrix is a *N* x *N* symmetric matrix defined from a data set with *N* input objects whose (*i,j*)th element represents the similarity or dissimilarity between the *i*th and *j*th objects.

The result of the clustering is a hierarchical classification of the objects following a taxonomy of is-a relations (i.e. class-subclass), known as *dendogram*. So objects belong to a set of nested clusters. A dendogram can be cut at a desired dissimilarity level obtaining a partition of the objects in disjoint classes. It is usual to perform the cut at the level that optimizes the Calinski-Harabasz index that maximizes the ratio between the inertia inter and intra clusters (Calinski and Harabasz, 1974).

HC algorithms are mainly classified based on the way of constructing the dendogram as *agglomerative clustering* and *divisive clustering*.

2.2.2.1 Agglomerative clustering

Agglomerative clustering starts with *N* clusters each of them including exactly one object and then a series of merge operations are followed out to construct a cluster including all individuals. This follows a bottom-up approach.

This type of clustering is the most used and fulfills the properties of sequentiality and exclusivity, also known as SAHN (P. H. A. Sneath and Sokal, 1973) (*Sequential, Agglomerative, Hierarchic and Nonoverlapping*).

The general agglomerative clustering can be summarized by the following procedure:

Step 1) Start with *N* singleton clusters and calculate the proximity matrix for the *N* clusters.

Step 2) Search the minimal distance in the proximity matrix between each pair of clusters and combine the pair of more similar clusters to form a new cluster.

Step 3) Update the proximity matrix by computing the distances between the new cluster and the other clusters to reflect this merge operation.

Step 4) Repeat steps 2)–3) until all objects are in the same cluster.

Now, we present different agglomerative clustering algorithms based on different ways of computing the proximity between clusters. The simplest and most popular methods are:

- *Single linkage* (P. Sneath, 1957): the distance between two clusters is determined as the minimum of the distances between all pairs of objects in different clusters. This method produces a reduction of the objects' space since it is taking the minimum distance at each step. One interesting consequence is that small changes between a pair of objects do not significantly modify the dendogram, meaning that the process is non sensitive to small variations. Single-linkage is sensitive to noise and outliers. Its tendency is to produce straggly or elongated clusters.
- *Complete linkage* technique (Sorensen, 1948): the distance between two clusters is determined as the maximum of all pairwise distances objects in the two clusters. This approach produces an expansion of the object's space. Complete linkage is less susceptible to outliers and noise. An interesting property is that it can break large clusters and produces compact clusters (Baeza-Yates, 1992). On the contrary to Single Linkage, this method is conservative because all pairs of objects must be related before the objects can form a cluster. In general this algorithm produces more useful hierarchies in many applications than Single linkage (Anil K. Jain and Dubes, 1988).
- *Average linkage:* the distance between two clusters is computed as the average of the distance among all the objects of the two clusters. The average can be calculated in different ways, but the most common is to use the arithmetic mean. Another version weights each object according to the number of elements of the cluster to which it belongs. In this case it is called "*group average*"(Sokal and Michener, 1958). There are other ways to assign weights to objects, such as depending on how the objects have been successively incorporated to the cluster.
- *Centroid linkage:* this approach considers also an artificial object that is built as the prototype of a cluster. The distance between two clusters is defined as the distance between their centroids. The centroid is calculated using some averaging function on each of the attributes that describe the objects.
- *Median linkage*: the distance between two clusters is based on an artificial point that is taken as the median of the two points that are creating the new cluster (Gower, 1967). This solves a drawback of the centroid approach, because if two clusters with very different size are fused, the centroids will have different degrees of representativeness with respect to their clusters. Considering that the centroid of the new cluster will lie along the median of the triangle defined by the clusters that are forming a new group and an external one, the median is proposed for the similarity computation.

- *Minimum-variance* or *Ward's method*: (Ward, 1963): the proximity between a pair of clusters is defined as the increase in the square error that results when two clusters are merged. This method attempts to minimize the sum of the square distances of objects with respect to the cluster prototype. In this way, the information loss, defined in terms of within-groups sum-of-squares, is minimized. Thus, minimizing the inertia inter-class, we are able to obtain a more optimum partition of the objects. It is worth to note that, if the function used to measure the distances is a metric (i.e. it fulfills the triangle inequality), the Huygens theorem of decomposition of inertia holds (Dillon and Goldstein, 1984). This property is related with the interpretability of the final clusters.

The error sum of squares of a cluster C_i when we have K variables is computed as:

$$ESS(C_i) = \sum_{j \in C_i} \sum_{l=1}^{K} \left(x_{jl} - \bar{x}_l^{(i)} \right)^2$$

where $\bar{x}^{(i)}$ is the vector of averages of the elements that belong to the cluster C_i.

Those six methods presented until now are the most well-known hierarchical clustering techniques. In spite of their differences, they share a common way of construction of the dendogram. Lance and Williams (Lance and Williams, 1967) proposed a parameterized updating formula to calculate distances between a new cluster and existing points, based on the distances prior to forming the new cluster. This recursive approach avoids the recalculation of the distances with respect to all the objects that belong to the clusters that are compared. This formula has three parameters, and each of the clustering methods can be characterized by its own set of Lance-Williams parameters (see Table 1).

Using the notation of Lance-Williams, let d_{ij} be the distance between points i and j and let $d_{k(ij)}$ be the updated distance of point k to the newly formed cluster (ij). Thus, d_{ij} is a within cluster distance and $d_{k(ij)}$ becomes a distance between clusters. The recursive formula is defined as:

$$d_{k(ij)} = \alpha_i d_{ki} + \alpha j d_{kj} + \beta d_{ij} + \lambda \mid d_{ki} - d_{kj} \mid$$

The α, β and λ variables are the parameters that define the linkage process. The following table shows the values of these parameters for the methods presented before. One feature of the recurrence formula is that any hierarchical clustering scheme which satisfies the relation will also possess a unique set of parameter values.

Table 1. Hierarchical Algorithms

Method	α_i	α_j	β	λ	Monotonic/Ultrametric
Single Linkage	1/2	1/2	0	-1/2	Yes
Complete Linkage	1/2	1/2	0	1/2	Yes
Group Average	$n_i/(n_i+n_j)$	$n_i/(n_i+n_j)$	0	0	Yes
Arithmetic Average	1/2	1/2	0	0	Yes
Centroid	$n_i/(n_i+n_j)$	$n_i/(n_i+n_j)$	$-n_in_j/(n_i+n_j)^2$	0	No
Median	1/2	1/2	-1/4	0	No
Minimum Variance (Ward)	$(n_i+n_k)/(n_i+n_j+n_k)$	$(n_j+n_k)/(n_i+n_j+n_k)$	$-n_k/(n_i+n_j+n_k)$	0	Yes

The n_i values refer to the number of elements in cluster i.

Agglomerative clustering methods can also be divided according to the way of representing the clusters:

- *graph methods*: consider all points of a pair of clusters when calculating their inter-cluster distance.
- *geometric methods*: use geometric centers of the clusters in order to determine the distance between them.

Table 2. Classification of the hierarchical methods presented in this section.

Graph methods	Single linkage, complete linkage and average linkage.
Geometric methods	Centroid linkage, median linkage and Ward's method

In recent years, with the requirement for handling large-scale data, new hierarchical techniques have appeared with the aim to minimize the computational cost of the classical algorithms. Some examples include:

- *BIRCH* (Zhang et al., 1997) (Balanced Iterative Reducing and Clustering using Hierarchies) is an incremental and hierarchical clustering algorithm for very large databases. The two main building components in the Birch algorithm are a hierarchical clustering component, and a main memory structure component. Birch uses a **main memory** (of limited size) data structure called *CF tree*. The tree is organized in such a way that (i) the leafs contain actual clusters, and (ii) the size of any cluster in a leaf is not larger than *R*. Initially, the data points are in one cluster. As the data arrives, a check is made whether the size of the cluster does not exceed *R*. If the cluster size grows too big, the cluster is split into two clusters, and the points are redistributed. The points are then continuously inserted to the cluster which enlarges less. At each node of the tree the CF tree keeps information about the mean of the cluster, and the mean of the sum of squares to compute the size of the clusters efficiently. The tree structure also depends on the branching parameter *T*, which determines the maximum number of children each node can have.
- *CURE* (Clustering Using REpresentatives)(Guha et al., 2001): it represents a cluster by a fixed number *h* of points scattered around it. The distance between two clusters used in the agglomerative process is equal to the minimum of distances between two scattered representatives. Therefore, CURE takes a middle-ground approach between the graph (all-points) methods and the geometric (one centroid) methods. CURE is capable of finding clusters of different shapes and sizes, and it is insensitive to outliers. CURE was designed to work with numerical values.
- *ROCK (Guha et al., 2000)* (The RObust Clustering using linKs) clustering algorithm is based on links between data points, instead of distances when it merges clusters. These links represent the relation between a pair of objects and their common neighbours. The notion of links between data helps to overcome the problems with distance based coefficients. For this reason, this method is extended to non-metric similarity measures that are relevant in situations where a domain expert/similarity table is the only source of knowledge. ROCK works with categorical features.
- *RCH (* Relative hierarchical clustering) considers both the internal distance (distance between a pair of clusters which may be merged to yield a new cluster) and the external distance (distance from the two clusters to the rest), and uses their ratio to decide the proximities (Mollineda and Vidal, 2000).
- SBAC (similarity-based agglomerative clustering) which was developed by Li and Biswas (C. Li and Biswas, 1999) extends agglomerative clustering techniques to deal with both numeric and nominal data. It employs a mixed data measure scheme that pays extra attention to less common matches of feature values (C. Li and Biswas, 2002).
- *CHAMELEON* (Karypis et al., 1999). It uses dynamic modelling in cluster aggregation. It uses a connectivity graph corresponding to the K-nearest neighbour model of sparsification of the proximity matrix, so that the edges of the k most similar points to any given point are preserved, and the rest are

pruned. CHAMELEON has two stages. In the first stage small tight clusters are built to ignite the second stage. In the second stage an agglomerative process is performed.

2.2.2.2 Divisive clustering

The divisive approach proceeds in a top-down manner. Initially, the entire data set belongs to a unique cluster and a procedure successively divides it until all clusters are singleton clusters.

The crucial points of this type of clustering are (1) the definition of a coherence function in order to select the next cluster to split, and (2) the definition of the splitting function. The former can be resolved by calculating the variance in the cluster and selecting the cluster with the highest variance for splitting, or simply detecting the largest cluster for splitting. About the latter, the splitting task usually consists on putting the data points into two different clusters. This type of methods has been less exploited because the agglomerative approach is more efficient.

Some divisive clustering algorithms are:

- *Bi-Section-Kmeans*: this clustering algorithm is an extension of the basic k-means, that divides one cluster in two (for k=2) at each step. The process ends when the desired number of clusters has been generated.
- *DIANA(DIvisive ANAlysis)*(Kaufman and Rousseeuw, 1990): is a heuristic method that consists on considering only a part of all the possible divisions at each step. Consists of a series of iterative steps to move the objects to the closest splinter. The splinter is initialized with the object that is farthest from the others.
- *MONA* (monothetic analysis) (Kaufman and Rousseeuw, 1990). When all the features are used together the algorithm is called polythetic. Otherwise, it is called monothetic, because only one feature is considered at each step. In (Kaufman and Rousseeuw, 1990) this approach is used with binary features, where the similarity is computed through association measures.
- *DIVCLUS-T* (Chavent et al., 2007) is a divisive hierarchical clustering algorithm based on a monothetic bipartitional approach allowing the dendrogram of the hierarchy to be read as a decision tree. It is designed for either numerical or categorical data. Like the Ward algorithm, it is based on the minimization of the inertia criterion. However, it provides a simple and natural interpretation of the clusters.

2.2.3 Other Clustering techniques

In addition to the hierarchical and partitional methods there are other clustering approaches (a good survey is done in the book of Xu et al. (Xu et al. 2009)). Here we present a summarized list of those techniques:

- *Densities-Based Clustering*: Here a cluster is understood as a dense region of objects that is surrounded by a region of low density. A known algorithm of this type is DBSCAN (Ester et al., 1996), which assigns the points that are close enough in the same cluster. Likewise, any border point that is close enough to a core point is put to the same cluster as the core point. However, noisy points are discarded producing a non complete clustering. Others are GMDD (Gaussian mixture density decomposition (GMDD) and AutoClass.

- *Model-based methods*: This approach is based on building an explicit model of each cluster (e.g. using a simple distribution function). The model determines which data belong to each cluster. This means that each cluster is considered as a model that can be described intrinsically, rather than as a collection of points assigned to it. A popular method for categorical data is COBWEB (Fisher, 1987). It uses incremental learning instead of following divisive or agglomerative approaches.

- *Neural Networks-Based Clustering*: Here objects are represented as neurons, these neurons increases the neighbourhood in some regions creating clusters and decrease it with other neurons. Some examples of this kind of algorithms are LVQ (Learning vector quantization), SOFMs (Self-Organized Feature Maps) and ART (Adaptative Resonance Theory).

- *Graph Theory-Based Clustering*: Here the data are represented as a graph where the nodes are objects and the links represent connections between objects. Then a cluster is defined as a group of objects that are connected between them but that have not connections with objects outside the group. A well-known graph-theoretic divisive clustering algorithm is based on the construction of the *minimal spanning tree* (MST) of the data (Zahn, 1971), and then deleting the MST edges with the largest lengths to generate clusters.

- *Kernel-based clustering*: The basis of this approach is that with a nonlinear transformation of a set of objects into a higher-dimensional space, one can find easily a linear separation of these objects into clusters. So, the goal is to change the space of representation of the objects. However, building a nonlinear mapping in the transformed space is usually a time-consuming task. This process can be avoided by calculating an appropriate inner-product kernel. The most common kernel functions include polynomial kernels and Gaussian radial basis functions (RBFs) and sigmoid kernels (Corchado and Fyfe, 2000).

- *Fuzzy clustering*: while traditional clustering approaches generate partitions where each data object belongs to one and only one cluster, *fuzzy* clustering extends this notion to associate each data object with every cluster using a membership function (Zadeh, 1965). Larger membership values indicate higher confidence in the assignment of the object to the cluster. The most widely used algorithm is the Fuzzy C-Means (FCM) algorithm (Sato et al., 1997), which is based on k-means. FCM attempts to find the most characteristic point in each cluster, which can be considered as the "center" of the cluster and, then, the grade of membership of each instance to the clusters. Many extensions of FCM are still being developed, such as (Hamasuna et al., 2010).

2.3 Discussion

In this section we make an analysis of the different clustering methods introduced in this document, focusing mainly in hierarchical and partitional techniques. Several observations are commonly done about

these techniques, which are really relevant for the user in order to select the appropriate methodology for a particular problem.

The first concern is about the fact that the selection of the clustering algorithm determines some characteristics of the clusters that are obtained. For example, center-based algorithms as the k-means will produce compact and spherical groups. Hierarchical methods organize groups on a multi-level groups and subgroups structure, which can be interesting to have in some particular applications. If a partition is generated from the dendrogram, then different types of clusters are obtained (see details in section 2.2.2). Other characteristics are obtained in density classification methods, which form groups according to the objects density; therefore, they do not limit the size of the group and very heterogeneous forms of groups can be found. If the user has some knowledge about the form of the clusters, then the selection of the clustering algorithm must be done according to this knowledge. For example, the single-link clustering algorithm works well on data sets containing non-isotropic clusters including well-separated, chain-like, and concentric clusters, whereas a typical partitional algorithm such as *k*-means works well only on data sets having isotropic clusters (Nagy, 1968).

If there is no information about the form of the clusters, one may consider the advantages and disadvantages of each of the different techniques. On the one hand, hierarchical algorithms are more versatile than partitional algorithms. The hierarchical representation provides very informative descriptions and visualization for the potential data clustering structures. On the other hand, the time and space complexities of the partitional algorithms are typically lower than those of the hierarchical algorithms(Day, 1992). In particular, partitional methods have advantages in applications involving large data sets for which the construction of a dendrogram is computationally prohibitive (A. K. Jain et al., 1999) (Everitt et al., 2001).

The main drawback of partitional algorithms is how to make the choice of the number of desired output clusters. In the case of hierarchical methods, they do not require the number of clusters to be known in advance as the final clustering results are obtained by cutting the dendrogram at different levels. However, their main disadvantage is that they suffer from their inability to perform adjustments once the splitting or merging decision is made (i.e. once an object is assigned to a cluster, it will not be considered again), which means that hierarchical algorithms are not capable of correcting a possible previous misclassification. This rigidity is useful because it leads to a smaller computational cost, since it does not have to worry for a combinatorial number of possible options. Moreover, most of the hierarchical methods have the tendency to form spherical shapes and the reversal phenomenon, in which the normal hierarchical structure is distorted.

Partitional algorithms perform a search on the space of features of the objects. So, they suffer from the problem of getting trapped in a local optimum and therefore being dependent on the initialization. Some approaches to find a global optimum introduce additional parameters, for which there are no theoretical guidelines to select the most effective value.

It can be observed that both approaches have advantages and disadvantages in different aspects. In that sense, it is possible to develop hybrid algorithms that exploit the good features of both categories (Murty and Krishna, 1980). Finally, in crisp clustering methods, clusters are not always well-separated. Fuzzy clustering overcomes the limitations of hard classification. However, a problem with fuzzy clustering is that it is difficult to define the membership values of the objects.

As a final and general remark, one can observe that there is not a best algorithm for all the cases. Depending on the purpose of the clustering, the most suitable approach must be selected.

3 Software tools for clustering

Different software tools are available, some of them developed by commercial companies:

Table 3. Data mining software tools

Name	SAS Enterprise Miner
Link	http://www.sas.com/technologies/analytics/datamining/miner/
Free	No
Description	Is a software tool able to perform data mining processes based on analysis of vast amounts of data with a broad set of tools. SAS provides a variety of clustering algorithms. It provides the different hierarchical agglomerative algorithms (single linkage, average linkage, complete linkage, centroid linkage or Ward's method). Also, it provides the K-Means algorithm, and the Self-Organizing Maps (SOM) algorithm (Kohonen Networks). It provides a wide range of graphic tools in order to study the results.
Name	SPSS
Link	http://www.spss.com/
Free	No
Description	SPSS is a powerful statistical tool and is one of the most widely used programs for statistical analysis in social science. It include descriptive statistics (Cross tabulation, Frequencies, Descriptives, Explore, Descriptive Ratio Statistics), bivariate statistics (Means, t-test, ANOVA, Correlation (bivariate, partial, distances), Nonparametric tests), Prediction for numerical outcomes (Linear regression), Prediction for identifying groups (Factor analysis, cluster analysis (two-step, K-means, hierarchical), Discriminant). It implements different clustering methods such as average linkage, single linkage, complete linkage, centroid method, median method, Ward's method and K-means.
Name	Clementine from SPSS
Link	http://www.spss.com/la/productos/clementine/clementine.htm
Free	No
Description	SPSS Clementine provides two clustering algorithms, which are the K-Means algorithm, and the Self-Organizing Maps algorithm (Kohonen Networks). SPSS Clementine cannot cluster data hierarchically and it cannot cluster data set that has categorical variables. However, it can cluster a data set specifying the number of clusters before the process using K-Means algorithm. Also, it can cluster a data set without specifying the number of clusters before the process using the Kohonen Network algorithm.
Name	Intelligent Miner (IBM)
Link	http://www-306.ibm.com/software/data/iminer/
Free	No
Description	IBM Intelligent Miner is a set of "statistical, processing, and mining functions" to analyze data. It contains three main products: Intelligent Miner Modeling, Intelligent Miner Scoring, and Intelligent Miner Visualization. The first one develops analytic models such are Associations, Clustering, Decision trees, and Transform Regression PMML models via SQL API. The second one performs scoring operation for the models created by Intelligent Miner Modeling. The last one presents data modeling results using one of the following Visualizers: Associations Visualizer, Classification Visualizer, Clustering Visualizer, and Regression Visualizer. IBM's Intelligent Miner provides a variety of data mining techniques: Predictive modeling, Database segmentation or clustering, Link analysis (associations), Neural Classification, Neural Clustering, Sequential Patterns, Similar Sequences, Radial Basis Function (RBF)-Prediction, and

	Deviation detection (outliers). In particular, it provides only two clustering algorithms: the Demographic algorithm, and the Self-Organizing Maps algorithm (Kohonen Networks). So, IBM DB2 cannot cluster data set hierarchically and cannot cluster a data set based on a predefined number of clusters. However it can cluster a data set that has categorical variables using the Demographic algorithm.
Name Link Free Description	WEKA (Waikato Environment for Knowledge Analysis) http://www.cs.waikato.ac.nz/ml/weka/ Yes Developed in the university of Waikato, New Zealand, Weka is a collection of machine learning algorithms for data mining tasks, implemented in Java. This software is one of the most complete ones of those free software packages. It can be executed from a command-line environment, or from a graphical interface, or it can be called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization, and is well-suited for developing new machine learning schemes. In particular, it contains different algorithms of clustering such as Cobweb, DBScan (Density-Based Spatial Clustering of Applications with Noise), EM (Expectation-Maximisation), Farthest-First, FilteredClusterer, OPTICS (Ordering Points To Identify the Clustering Structure), x-means, MakeDensityBased-Clusterer algorithm, SimpleKMeans, CLOPE, SiB (sequential Information Bottleneck).
Name Link Free Description	Pentaho http://www.pentaho.com/ No Pentaho Data Mining, provides a comprehensive set of machine learning algorithms from Weka. Its broad suite of classification, regression, association rules, segmentation, decision trees, random forests, neural networks, and clustering algorithms can be used to help an analyst understand the business better and to improve future performance through predictive analytics. So it have the same clustering algorithms than weka.
Name Link Free Description	RapidMiner http://rapid-i.com/ Yes RapidMiner (formerly YALE (Yet Another Learning Environment)) is an open source toolkit for data mining. RapidMiner implements different clustering methods as, DBScan, EM, the Weka clustering schemes, Kernel K-Means, K-Means, K-Medoids, a Random Clustering, and an implementation of Support Vector Clustering.
Name Link Free Description	R language http://www.r-project.org/ Yes R is a programming language and environment for statistical computing and graphics. Available for Windows, various Unix flavors (including Linux), and Mac. Provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. And it has interoperability with other languages as C, XML and Java. A number of different clustering methods are provided in this software. *Ward's* minimum variance method aims at finding compact, spherical clusters. The *complete linkage* method finds similar clusters. The *single linkage* method adopts a 'friends of friends' clustering strategy. The other methods can be regarded as aiming for clusters with characteristics somewhere between the single and complete link methods. Note however, that methods "median" and "centroid" are *not* leading to a *monotone distance* measure, or equivalently the resulting dendrograms can have so

	called *inversions* (which are hard to interpret).
Name Link Free Descrip-tion	Rattle, Gnome Cross Platform GUI for Data Mining using R http://rattle.togaware.com/ Yes Rattle(R Analytical Tool To Learn Easily) is a data mining toolkit used to analyze very large collections of data. Rattle presents statistical and visual summaries of data, transforms data into forms that can be readily modeled, builds both unsupervised and supervised models from the data, presents the performance of models graphically, and scores new datasets. It has a simple and logical graphical user interface based on Gnome. Rattle runs under GNU/Linux, Macintosh OS/X, and MS/Windows. In addition, Rattle can be used by itself to deliver data mining projects. Rattle also provides an entry into sophisticated data mining using the open source and free statistical language R.
Name Link Free Descrip-tion	Tanagra http://chirouble.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html Yes It is a free (open-source) data-mining package that contains components for Data source (tab-delimited text), Visualization (grid, scatterplots), Descriptive statistics (cross-tab, ANOVA, correlation), Instance selection (sampling, stratified), Feature selection and construction, Regression (multiple linear), Factorial analysis (principal components, multiple correspondence), Clustering, Supervised learning (logistic regr., k-NN, multi-layer perceptron, prototype-NN, ID3, discriminant analysis, naive Bayes, radial basis function), Meta-spv learning (instance Spv, arcing, boosting, bagging), Learning assessment (train-test, cross-validation), and Association (Agrawal a-priori). It provides different clustering methods such as kMeans, Kohonen's Self Organization Map, LVQ (Kohonen's Learning Vector Quantizers), a "supervised" clustering algorithm, and HAC (Hierarchical agglomerative clustering).
Name Link Free Descrip-tion	STATISTICA Data Miner http://www.statsoft.com/ No *STATISTICA Data Miner* contains a selection of data mining solutions, with an easy-to-use user interface and deployment engine. STATISTICA Data Miner is highly customizable and can be tailored to meet very specific and demanding analysis requirements through its open architecture. Some characteristics are machine Learning (Bayesian, Support Vectors, Nearest Neighbour), General Classification/Regression tree models, General CHAID models,Boosted Tree Classifiers and Regression, Random Forests for Regression and Classification, MARSplines (Multivariate Adaptive Regression Splines), Cluster Analysis, Combining Groups (Classes) for Predictive Data Mining, Automatic Feature Selection, Ensembles of Neural Networks, etc. It provides different clustering methods such as kMeans and Generalized EM.
Name Link Free Descrip-tion	*CLUTO* http://glaros.dtc.umn.edu/gkhome/views/cluto/ Yes CLUTO is a family of data clustering and cluster analysis programs and libraries, that are well suited for low- and high-dimensional data sets. CLUTO is well-suited for clustering data sets arising in many diverse application areas including information retrieval, customer purchasing transactions, web, GIS, science, and biology. It has multiple classes of clustering algorithms (partitional, agglomerative (single-link, complete-link, UPGMA), and graph-partitioning based) and multiple similarity/distance functions (Euclid-

	ean distance, cosine, correlation coefficient, extended Jaccard, user-defined).
Name	Oracle Data Mining (ODM)
Link	http://www.oracle.com/technology/products/bi/odm/index.html
Free	NO
Description	Oracle Data Mining is an option of Oracle Corporation's Relational Database Management System (RDBMS) Enterprise Edition (EE). It contains several data mining and data analysis algorithms for classification, prediction, regression, clustering, associations, feature selection, anomaly detection, feature extraction, and specialized analytics. ODM offers well known machine learning approaches such as Decision Trees, Naive Bayes, Support vector machines, Generalized linear model (GLM) for predictive mining, Association rules, K-means (Enhanced k-means (EKM)) and Orthogonal Partitioning Clustering (O-Cluster), and Non-negative matrix factorization for descriptive mining.
Name	DBMiner
Link	http://www.pentaho.com/
Free	No
Description	DBMiner implements a wide spectrum of data mining functions, including generalization, characterization, association, classification, and prediction. By incorporating several interesting data mining techniques, including attribute-oriented induction, statistical analysis, progressive deepening for mining multiple-level knowledge, and meta-rule guided mining, the system provides a user-friendly, interactive data mining environment with good performance.The underlying algorithm used in DBMoner is the *k*-means method

4 Semantic similarity measures

With the enormous success of the Information Society and the World Wide Web, the amount of textual electronic information available has significantly increased. As a result, computer understanding of text has acquired great interest in the research community in order to enable a proper exploitation, management, classification or retrieval of textual data.

One of the most basic problems when aiming to interpret textual data is the assessment of semantic likeness between words because, as it has been demonstrated in psychological experiments (Goldstone, 1994), it acts as a fundamental organizing principle by which humans organize and classify objects. It is important to note that two different concepts, which are often confounded, can be found in the literature. On one hand, *semantic similarity* states how taxonomically near two terms are, because they share some aspects of their meaning (*e.g.*, *dogs* and *cats* are similar to the extent they are mammals). On the other hand, the more general concept of *semantic relatedness* does not necessarily rely on a taxonomic relation (*e.g.*, *car* and *wheel* or *pencil* and *paper*); other non taxonomic relationships (*e.g.*, meronymy, antonymy, functionality, cause-effect, etc.) are also considered.

Semantic similarity/relatedness computation has many direct and relevant applications. Some basic natural language processing tasks such as word sense disambiguation (Patwardhan et al., 2003), synonym detection (Lin, 1998) or automatic spelling error detection and correction (Budanitsky and Hirst, 2001) rely on the assessment of words' semantic resemblance. Direct applications can be found in the knowledge management field, such as thesauri generation (Curran, 2002), information extraction (Stevenson and Greenwood, 2005) or ontology learning (Sánchez and Moreno, 2008), in which new terms related to already existing concepts, should be acquired from textual resources. The Semantic Web is an especially relevant application area, when dealing with automatic annotation of Web pages (Cimiano et al., 2004), community mining (Mika, 2007), and keyword extraction for inter-entity relation representation (Mori et al., 2007).

Similarity estimation between textual entities has also an important role in the classification and structuring of textual resources such as digital libraries (Sánchez and Moreno, 2007), in which resources should be classified according to the similarity of their main topics (expressed as textual signatures), and in information retrieval (IR), in which similar or related words can be used to expand user queries and improve recall (Sahami and Heilman, 2006). It is also exploited in the elaboration of methods for integrating the knowledge of different data bases into unique queries, where equivalent concepts must be identified (Schallehn et al., 2004).

Due to the proliferation of textual data referring to user descriptions (*e.g.*, polls or questionnaires), word similarity measurement can aid to develop specific data mining algorithms that take into account the semantics of the values. This is the case of clustering or classification techniques (Batet et al., 2008; Y. Chen et al., 2009) that can be used to detect user profiles and preferences, aiding the development of decision support systems.

Applied domains such as biomedicine, chemistry or engineering are especially considered by the research community (H. Al-Mubaid and Nguyen, 2006; Armengol, 2009; Hliaoutakis, 2005; Morbach et al., 2007; Pedersen et al., 2007; Pirró, 2009) due to the proliferation and importance of terminology. In this case, similarity assessment can aid to discover semantically equivalent terms corresponding to different lexicalizations, synonyms, abbreviations or acronyms of the same concept. This is of great interest in healthcare in order to be able to retrieve the desired information from a literature data base, especially tasks such as patient cohort identification (Bichindaritz and Akkineni, 2006; Pedersen et al., 2007).

Some video and image understanding techniques are also based on the semantic interpretation of the textual features referred to the images for indexing or searching purposes (Allampalli-Nagaraj and Bichindaritz, 2009). Semantic filtering of multimedia content needs to discover the relationships that exist between semantic concepts. In (Naphade and Huang, 2001), some relevant concepts may not be directly observed in terms of media features, but are inferred based on their semantic likeness with those that are already detected.

Despite its usefulness, robust measurement of semantic similarity/relatedness between textual terms remains a challenging task (Bollegala et al., 2007). Many works have been developed in the last years, especially with the increasing interest on the Semantic Web. Proposed methods aim to automatically assess a numerical score between a pair of terms according to the semantic evidence observed in one or several knowledge sources, which are used as semantic background. According to the concrete knowledge sources exploited for the semantic assessment (*e.g.*, ontologies, thesaurus, domain corpora, etc.) and the way to use them, different families of methods can be identified.

According to the corpus exploited to extract semantic evidences and the principles in which similarity estimation relies, measures can be grouped in several families of functions. In this section, we survey, review and compare them according to the following classification:

1. Ontology-based measures relying on:
    - 1.1. Edge-counting
    - 1.2. Features
    - 1.3. Information Content (corpora-dependent or intrinsic to an ontology)
2. Distributional measures based on:
    - 2.1. First-order co-occurrence
    - 2.2. Second-order co-occurrence (relying on corpora or on structured thesaurus glosses)

4.1 Ontology-based measures

Ontologies provide a formal specification of a shared conceptualization (Guarino, 1998). Being machine readable and constructed from the consensus of a community of users or domain experts, they represent a very reliable and structured knowledge source. Due to this reason, and thanks to initiatives such as the Semantic Web, which brought the creation of thousands of domain ontologies (Ding et al., 2004), ontologies have been extensively exploited to compute semantic likeness.

A paradigmatic example is WordNet (Fellbaum, 1998), a domain-independent and general purpose ontology/thesaurus that describes and organizes more than 100,000 general English concepts, which are semantically structured in an ontological fashion. It contains words (nouns, verbs, adjectives and adverbs) that are linked to sets of cognitive synonyms (*synsets*), each expressing a distinct concept (*i.e.,* a word sense). Synsets are linked by means of conceptual-semantic and lexical relations such as synonymy, hypernymy (is-a), six types of meronymy (part-of), antonymy, complementary, etc. The backbone of the network of words is the subsumption hierarchy which accounts for more than 80% of all the modelled semantic links, with a maximum depth of 16 nodes. The result is a network of meaningfully related words, where the graph model can be exploited to interpret the meaning of the concept.

In this section, we cover approaches completely or partially relying on ontologies to compute semantic similarity/relatedness. WordNet has been mainly used as the background ontology.

4.1.1 Edge counting-based measures

Ontologies can be seen as a directed graph in which concepts are interrelated mainly by means of taxonomic (is-a) and, in some cases, non-taxonomic links. Input terms are mapped to ontological concepts by means of their textual labels. A straightforward method to calculate the similarity between terms is to evaluate the minimum *Path Length* connecting their corresponding ontological nodes via is-a links (Rada et al., 1989). The longer the path, the more semantically far the terms are.

Let us define $path(a,b)=l_1,....l_k$ as a set of links connecting the terms a and b in a taxonomy. Let $|path(a,b)|=k$ be the length of this path. Then, considering all the possible paths from a to b, their semantic distance as defined by (Rada et al., 1989) is (1).

$$dis_{rad}(a,b) = \min_{\forall i} |path_i(a,b)| \quad (1)$$

Several variations and improvements of this edge-counting approach have been proposed. On one hand, in addition to this absolute distance between terms, Wu and Palmer (Wu and Palmer, 1994) considered that the relative depth in the taxonomy of the concepts corresponding to the evaluated terms is an important dimension, because concept specializations become less distinct as long as they are recursively refined. So, equally distant pairs of concepts belonging to an upper level of a taxonomy should be considered less similar than those belonging to a lower lever. Wu and Palmer's measure counts the number of is-a links (N_1 and N_2)

from each term to their Least Common Subsumer (LCS) (*i.e.,* the most concrete taxonomical ancestor that subsumes both terms) and also the number of is-a links of the LCS to the root (N_3) of the ontology (2).

$$sim_{w\&p}(a,b)\frac{2\times N_3}{N_1+N_2+2\times N_3} \quad (2)$$

Based on the same principle, Leadcock and Chodorow (Leacock and Chodorow, 1998) also proposed a measure that considers both the number of nodes N_p separating the ontological nodes corresponding to terms *a* and *b*, included themselves, and the depth *D* of the taxonomy in which they occur in a non-linear fashion (3).

$$sim_{l\&c}(a,b) = -\log(N_p/2D) \quad (3)$$

Li *et al*., (Y. Li et al., 2003) also proposed a similarity measure that combines the shortest path length and the depth of ontology information in a non-linear function (4).

$$sim_{li}(a,b) \ = e^{-\alpha \min_{\forall i}|path_i(a,b)|}.\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}} \quad (4)$$

, where *h* is the minimum depth of the LCS in the hierarchy and $\alpha \geq 0$ and $\beta > 0$ are parameters scaling the contribution of the shortest path length and depth, respectively. Based on benchmark data, authors stated that the optimal parameters for the measure with respect to a concrete set of human judgements were: $\alpha = 0.2$; $\beta = 0.6$. However, this is just an empirical finding for a specific setting. It lacks a theoretical basis and cannot be generalized.

Al-Mubaid and Nguyen (H. Al-Mubaid and Nguyen, 2006) proposed a cluster-based measure that combines the minimum *path length* and the *taxonomical depth.* They define clusters for each of the branches in the hierarchy with respect the root node. They measure the common specificity of two terms by substracting the depth of their LCS from the depth D_c of the cluster (5).

$$CSpec(a,b) = D_c - depth(LCS(a,b)) \quad (5)$$

The common specificity is used to consider that lower level pairs of concept nodes are more similar than higher level pairs, as in Wu and Palmer's approach. So, the proposed distance measure (*sem*) is defined as follows (6):

$$dis_{sem}(a,b) = \log((\min_{\forall i}|path_i(a,b)|-1)^{\alpha}\times(CSpec)^{\beta}+k) \quad (6)$$

, where $\alpha > 0$ and $\beta > 0$ are the contribution factors of the path length and the common specify features and *k* is a constant. Authors use $k=1$ because with $k \geq 1$ they proved that the distance is positive. Moreover, in their experiments, they give an equal weight to the contribution of the two components (path length and common specify) by using $\alpha = \beta = 1$.

Both Li *et al*., and Al-Mubaid and Nguyen approaches are often considered in the literature (Petrakis et al., 2006; Pirró, 2009) as "hybrid" approaches, as they combine several structural characteristics (such as path length, depth and local density) and assign weights to balance the contribution of each component to the final similarity value. Even though their accuracy for a concrete scenario (see evaluation section) is higher than more basic edge-counting measures, they depend on the empirical tuning of weights according to the ontology and input terms.

Hirst and St-Onge (Hirst and St-Onge, 1998) extended the notion of taxonomical edge-counting by considering also non-taxonomic semantic links in the path (*full_path*). All types of relations found in WordNet together with rules that restrict possible semantic chains are considered, along with the intuition that the longer the path and the more changes in relation's direction, the lower the likeness. The following path directions are considered: upward (such as *hypernymy* and *meronymy*), downward (such as *hyponymy* and *holonymy*) and horizontal (such as *antonymy*). The resulting formula is (7)

$$sim_{h\&s}(a,b) = C - full\_path(a,b) - k \times turns(a,b) \qquad (7)$$

, where C and k are constants (C = 8 and k = 1 are used by the authors), and *turns(a, b)* is the number of times the path's direction changes.

Due to the non-taxonomic nature of some of the relations considered during the assessment, Hirst and St-Onge's measure captures a more general sense of *relatedness* than of taxonomical *similarity,* assessed by the approaches detailed above.

The main advantage of the presented measures is their simplicity. They only rely on the geometrical model of an input ontology whose evaluation requires a low computational cost (in comparison to approaches dealing with text corpora, see Section 4.2). However, several limitations hamper their performance.

In general, any ontology-based measure depends on the degree of completeness, homogeneity and coverage of the semantic links represented in the ontology. So, they require rich and consistent ontologies like WordNet to work properly (Pirró, 2009).

For the concrete case of taxonomic path-based measures, they only consider the shortest path between concept pairs. However, wide and detailed ontologies such as WordNet incorporate multiple taxonomical inheritance, resulting in several taxonomical paths which are not taken into account. Other features also influencing the concept semantics, such as the number and distribution of common and non-common taxonomical ancestors, are not considered either. As a result, by taking only the minimum path between concepts, many of the taxonomical knowledge explicitly modelled in the ontology is omitted.

Another problem of path-based measures typically admitted (Bollegala et al., 2009; Wan and Angryk, 2007) is that they rely on the notion that all links in the taxonomy represent a uniform distance. In practice, the semantic distance among concept specializations/generalizations in an ontology depends on the degree of granularity and taxonomic detail implemented by the knowledge engineer.

4.1.2 Feature-based measures

Feature-based methods try to overcome the limitations of path-based measures regarding the fact that taxonomical links in an ontology do not necessary represent uniform distances. This fact is addressed by considering the degree of likeness between sets of ontological features. As a result, they are more general and, potentially, they could be applied in cross ontology similarity estimation settings (*i.e.*, when concept pairs belong to two different ontologies), a situation in which edge-counting methods cannot be directly applied (Petrakis et al., 2006).

So, on the contrary to edge-counting measures which, as stated above, are based on the notion of minimum path distance, feature-based approaches assess similarity between concepts as a function of their properties. This is based on Tversky's model of similarity, which, derived from set theory, takes into account common and non common features of compared terms. Common features tend to increase similarity and non-common ones tend to diminish it (Tversky, 1977). Formally, let $\Psi(a)$ and $\Psi(b)$ be the features of terms *a* and *b* respectively, let $\Psi(a) \cap \Psi(b)$ be the intersection between those two sets of features, and $\Psi(a) \setminus \Psi(b)$ the set obtained when eliminating the elements of $\Psi(b)$ from the set of features of concept a, $\Psi(a)$. Then, the similarity between a and b can be computed as a function of $\Psi(a) \cap \Psi(b)$, $\Psi(a) \setminus \Psi(b)$ and $\Psi(b) \setminus \Psi(a)$ as (8).

$$sim(a,b) = \alpha \cdot F(\Psi(a) \cap \Psi(b)) - \beta \cdot F(\Psi(a) \setminus \Psi(b)) - \gamma \cdot F(\Psi(b) \setminus \Psi(a)) \qquad (8)$$

, where F is a function that reflects the salience of a set of features, and α, β and γ are parameters that weight the contribution of each component.

The information provided by the input ontology is exploited by the features. For WordNet, concept synonyms (*i.e.*, *synsets*, which are sets of linguistically equivalent words), definitions (*i.e.*, *glosses*, containing textual descriptions of word senses) and different kinds of semantic relationships can be considered.

In Rodriguez and Egenhofer (Rodríguez and Egenhofer, 2003), the similarity is computed as the weighted sum of similarities between synsets, meronyms and neighbour concepts (those linked via semantic pointers) of evaluated terms (9).

$$sim_{r\&e}(a,b) = w \cdot S_{synsets}(a,b) + u \cdot S_{meronyms}(a,b) + v \cdot S_{neighborhoods}(a,b) \qquad (9)$$

, where *w*, *u* and *v* weight the contribution of each component, which depends on the characteristics of the ontology. By meronyms, they evaluate matchings of concepts via part-of relationships.

In Tversky (Tversky, 1977) concepts and their neighbours (according to semantic pointers) are represented by synsets. The similarity (10) is computed as:

$$sim_{tve}(a,b) = \frac{|A \cap B|}{|A \cap B| + \gamma(a,b)|A \setminus B| + (1-\gamma(a,b))|B \setminus A|} \qquad (10)$$

, where A, B are the synsets for concepts corresponding to a and b, $A \backslash B$ is the set of terms in A but not in B and $B \backslash A$ the set of terms in B but not in A. Finally, $\gamma(a, b)$ is computed as a function of the depth of a and b in the taxonomy as follows (11):

$$\gamma(a,b) = \begin{cases} \dfrac{depth(a)}{depth(a)+depth(b)}, & depth(a) \le depth(b) \\ 1 - \dfrac{depth(a)}{depth(a)+depth(b)}, & depth(a) > depth(b) \end{cases} \quad (11)$$

In Petrakis *et al*., (Petrakis et al., 2006) a feature-based function called *X-similarity* relies on the matching between synsets and concept glosses extracted from WordNet (*i.e.*, words extracted by parsing term definitions). They consider that two terms are similar if the synsets and glosses of their concepts and those of the concepts in their neighbourhood (following semantic links) are lexically similar. The similarity function is expressed as follows:

$$sim_{X-Similarity}(a,b) = \begin{cases} 1, if\ S_{synsets}(a,b) > 0 \\ \max\{S_{neighborhoods}(a,b), S_{glosses}(a,b)\}, if\ S_{synsets}(a,b) = 0 \end{cases} \quad (12)$$

, where $S_{neighborhoods}$ is calculated as follows:

$$S_{neighborhoods}(a,b) = \max \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \quad (13)$$

, where each different semantic relation type (*i.e.*, *is-a* and *part-of* in WordNet) is computed separately (i denotes the relation type) and the maximum (joining all the synsets of all concepts up to the root of each hierarchy) is taken. $S_{glosses}$ and $S_{synsets}$ are both computed as:

$$S(a,b) = \max \frac{|A \cap B|}{|A \cup B|} \quad (14)$$

, where A and B denote synsets or glosses sets for terms a and b.

Feature-based measures exploit more semantic knowledge than edge-counting approaches, evaluating both commonalties and differences of compared concepts. However, by relying on features like glosses or synsets (in addition to taxonomic and non-taxonomic relationships), those measures limit their applicability to ontologies in which this information is available. Only big ontologies/thesaurus like WordNet include this kind of information. In fact, an investigation of the structure of existing ontologies via the Swoogle ontology search engine (Ding et al., 2004) reveals that domain ontologies very occasionally model any semantic feature apart from taxonomical relationships.

Another problem is their dependency on the weighting parameters that balance the contribution of each feature (like the *hybrid* approaches). In all cases, those parameters should be tuned according the nature of the ontology and even to the evaluated terms. This hampers their applicability as a general purpose solution.

Only Petrakis (Petrakis et al., 2006) does not depend on weighting parameters, as the maximum similarity provided by each single feature is taken. Even though this adapts the behaviour of the measure to the characteristics of the ontology and to the knowledge modelling, the contribution of other features is omitted if only the maximum value is taken at each time.

4.1.3 Information Content-based measures

Also acknowledging some of the limitations of edge-counting approaches, Resnik (Resnik, 1995) proposed to complement the taxonomical knowledge provided by an ontology with the information distribution of concepts evaluated in the input corpora. He exploited the notion of Information Content (IC), by associating appearance probabilities to each concept in the taxonomy, computed from their occurrences in a given corpus. The IC of a term *a* is computed as the inverse of its probability of occurrence, *p*(*a*) (15). In this manner, infrequent words are considered more informative than common ones.

$$IC(a) = -\log P(a) \tag{15}$$

According to Resnik, semantic similarity depends on the amount of shared information between two terms, a dimension which is represented by their Least Common Subsumer (LCS) in an ontology. Two terms are maximally dissimilar if a LCS does not exist (*i.e.*, in terms of edge-counting, it would not be possible to find a path connecting them). Otherwise, their similarity is computed as the IC of the LCS (16).

$$sim_{res}(a,b) = IC(LCS(a,b)) \tag{16}$$

One of the problems of Resnik's metric is that any pair of terms having the same LCS results in exactly the same semantic similarity. Both Lin (Lin, 1998) and Jiang and Conrath (Jiang and Conrath, 1997) extended Resnik's work by also considering the IC of each of the evaluated terms.

Lin enounced that the similarity between two terms should be measured as the ratio between the amount of information needed to state their commonality and the information needed to fully describe them. As a corollary of this theorem, his measure considers, on one hand, commonality in the same manner as Resnik's approach and, on the other hand, the IC of each concept alone (17).

$$sim_{lin}(a,b) = \frac{2 \times sim_{res}(a,b)}{(IC(a) + IC(b))} \tag{17}$$

The measure proposed by Jiang and Conrath is based on quantifying, in some way, the length of the taxonomical links as the difference between the IC of a concept and its subsumer. When comparing term pairs, they compute their distance by substracting the sum of the IC of each term alone from the IC of their LCS (18).

$$dis_{j\&c}(a,b) = (IC(a) + IC(b)) - 2 \times sim_{res}(a,b) \tag{18}$$

It is important to note that IC-based measures need, in order to behave properly, that the probability of appearance *p* monotonically increases as one moves up in the taxonomy (*i.e.*, $\forall c_i \mid c_j$ *is hypernym of* c_i => $p(c_i) \leq p(c_j)$). This is achieved by computing *p(a)* as the probability of encountering any *instance* of *a* in the given corpus. In practice, each individual occurrence of any noun in the corpus is counted as an occurrence of each taxonomic class containing it (19) (Resnik, 1995).

$$p(a) = \frac{\sum_{w \in W(a)} count(w)}{N} \qquad (19)$$

, where $W(a)$ is the set of nouns in the corpus whose senses are subsumed by *a,* and N is the total number of corpus nouns that are present in the taxonomy.

As a result, an accurate computation of concept probabilities requires a proper disambiguation and annotation of each noun found in the corpus. If either the taxonomy or the corpus changes, re-computations are needed to be recursively executed for the affected concepts. So, it is necessary to perform a manual and time-consuming analysis of corpora and resulting probabilities would depend on the size and nature of input corpora. Moreover, the background taxonomy must be as complete as possible (*i.e.*, it should include most of the specializations of each concept) in order to provide reliable results. Partial taxonomies with a limited scope may not be suitable for this purpose. All those aspects limit the scalability and applicability of those approaches.

Considering the limitations of IC-based approaches due to their dependency on corpora, some authors tried to intrinsically derive IC values from an ontology. Those works rely on the assumption that the taxonomic structure of ontologies like WordNet is organized in a meaningful way, according to the principle of *cognitive saliency* (Blank, 2003). This states that humans specialise concepts when they need to differentiate them from already existing ones. So, concepts with many hyponyms (*i.e.*, specializations) provide less information than the concepts at the leaves of the hierarchy. From the Information Theory point of view, they consider that abstract ontological concepts appear more probably in a corpus as they subsume many other ones. In this manner, they estimate the probability of appearance of a concept and in consequence, the amount of information that a concept provides, as a function of the number of hyponyms and/or their relative depth in the taxonomy.

Seco *et al*., (Seco et al., 2004) and Pirró and Seco (Pirró and Seco, 2008) base IC calculations on the number of hyponyms. Being *hypo(a)* the number of hyponyms of the concept *a* and *max_nodes* the maximum number of concepts in the taxonomy, they compute the IC of a concept in the following way (20):

$$IC_{seco}(a) = -\log(p(a)) = 1 - \frac{\log(hypo(a)+1)}{\log(max\_nodes)} \qquad (20)$$

The denominator (corresponding to the most informative concept) ensures that IC values are normalized in the range 0..1.

This approach only considers hyponyms of a given concept in the taxonomy; so, concepts with the same number of hyponyms but different degrees of generality appear to be equally similar. In order to tackle the problem, and in the same manner as for edge-counting measures, Zhou *et al*., (Zhou et al., 2008) proposed to complement the hyponym-based IC computation with the relative depth of each concept in the taxonomy. The IC of a concept is computed as (21):

$$IC_{zhou}(a) = k(1 - \frac{\log(hypo(a)+1)}{\log(max\_nodes)}) + (1-k)(\frac{\log(deep(a))}{\log(max\_depth)}) \quad (21)$$

In addition to *hypo* and *max_nodes*, which have the same meaning as eq. 20, *deep(a)* corresponds to the depth of the concept *a* in the taxonomy and *max_depth* is the maximum depth of the taxonomy. *K* is a factor that adjusts the weight of the two features involved in the IC assessment. They use k=0.5.

Both ways of computing IC intrinsically have been applied directly on the similarity functions proposed by Resnik, Lin and Jiang and Conrath. Those approaches overcome most of the problems observed for corpus-based IC approaches (specifically, the need of corpus processing and their high data-sparseness) competing and even improving with them in terms of accuracy (as it will be stated in the evaluation) when applied over WordNet. However, they require big, detailed and fine grained taxonomies/ontologies in order to enable an accurate estimation of the concept's IC. For small or very specialized ontologies with a limited taxonomical depth and low branching factor, the resulting IC values between concepts would be too homogenous to enable a proper differentiation.

4.2 Distributional approaches

On the contrary to ontology-based measures, distributional approaches only use text corpora as the source to infer semantics. They are based on the assumption that words with similar distributional properties have similar meanings (Waltinger et al., 2009); so, they infer semantic likeness from word co-occurrences in text corpora. As words may co-occur due to many different reasons (*i.e.,* being taxonomically related or not), distributional measures capture the more general sense of *relatedness* in contrast to taxonomically-based *similarity* measures.

According to the way in which distributional resemblance is determined, one may distinguish two different approaches. On one hand, some authors measure similarity from direct word co-occurrence in text (first order co-occurrence). On the other hand, other authors estimate relatedness as a function of the similarity of the contexts in which words occur (second order co-occurrence). In this section, we survey the main proposals of each kind.

4.2.1 First order co-occurrence

First order co-occurrence-based approaches rely on the principle that the more frequently two words appear together, the higher their relatedness. This follows the simple cognitive principle that people would judge two words as similar because they are exposed to them simultaneously (Lemaire and Denhière, 2006).

Being completely corpora-dependant, the choice of input data is crucial for these methods. In order to extract reliable conclusions, the corpus should be as representative as possible with regards to the real social-scale information distribution. For practical reasons, the analysis is restricted to textual sources, mainly due to the fact that people usually learn words from texts (Landauer and Dumais, 1997). As a result, the text corpora size and heterogeneity are important dimensions for being able to capture global-scale knowledge.

The Web, being the biggest electronic repository currently available, created from the interaction of a big community of users, represents one of the best options to apply those measures (Sánchez and Moreno, 2008). In fact, unsupervised models perform better when statistics are obtained from the Web rather than from other large corpora (Keller and Lapata, 2003).

First order approaches estimate relatedness as a function of the probability of co-occurrence of two terms in relation to individual probabilities. As computing absolute term appearances in the Web is very time consuming, authors associate probabilities to page counts provided by Web search engines. It is important to note that those engines estimate the number of appearances of a given query in individual documents but not the total amount of appearances (*e.g.*, in case of several appearances per document).

Pointwise Mutual Information (PMI) was one of the first functions to be adapted to the Web to compute term appearance probabilities from the Web page count (Turney, 2001). It is defined as the comparison between the probability of observing *a* and *b* together (estimated from the *page count* of the query '*a* AND *b'*) and observing them independently (estimated from the *page count* when querying *a* and *b* alone). If they are not statistically independent, they will have a tendency to co-occur (which is the case of words in a corpus) and the numerator will be greater than the denominator. Therefore, the resulting ratio (22) is a measure of the degree of statistical dependency between *a* and *b* (Turney, 2001). For the remainder of this document we use the notation *H(a)* and *H(b)* to denote the page count (*i.e.,* hits) provided by a search engine when querying '*a'* and '*b'* respectively and *H(a,b)* the page count when query '*a* AND *b'*. *M* is the total number of pages indexed by the search engine.

$$PMI(a,b) = -\log \frac{\frac{H(a,b)}{M}}{\frac{H(a)}{M}\frac{H(b)}{M}} \qquad (22)$$

Cilibrasi and Vitanyi (Cilibrasi and Vitányi, 2006), by carefully studying Information Theory, proposed a distance metric between words using Web search engine's page counts. It is defined as the normalized information distance (Y. Li et al., 2003) between two words. The function, named *Normalised Google Distance* (NGD) is defined as follows (23):

$$NGD(a,b) = \frac{\max[\log H(a), \log H(b)] - \log H(a,b)}{\log M - \min[\log H(a), \log H(b)]} \quad (23)$$

Bollegala (Bollegala et al., 2007) adapted several classical co-occurrence measures: Jaccard (24), Overlap (Simpson) (25), Dice (26) and the mentioned PMI in a similar way as Turney did (27).

$$WebJaccard(a,b) = \begin{cases} 0 & if\ H(a,b) \le \lambda \\ \dfrac{H(a,b)}{H(a) + H(b) - H(a,b)} & otherwise \end{cases} \quad (24)$$

$$WebOverlap(a,b) = \begin{cases} 0 & if\ H(a,b) \le \lambda \\ \dfrac{H(a,b)}{\min(H(a), H(b))} & otherwise \end{cases} \quad (25)$$

$$WebDice(a,b) = \begin{cases} 0 & if\ H(a,b) \le \lambda \\ \dfrac{2H(a,b)}{H(a) + H(b)} & otherwise \end{cases} \quad (26)$$

$$WebPMI(a,b) = \begin{cases} 0 & if\ H(a,b) \le \lambda \\ \log_2\left(\dfrac{\frac{H(a,b)}{M}}{\frac{H(a)}{M}\frac{H(b)}{M}}\right) & otherwise \end{cases} \quad (27)$$

In order to minimize the influence of noise in Web data, they set each coefficient to zero if the page count for the query *a* AND *b* is less than a threshold ($\lambda = 5$ was used in (Bollegala et al., 2007)). This omits some cases of random co-occurrences and misspelled terms. *M* is estimated as 10^{10} according to the number of indexed pages reported by Google in 2007.

Instead of using the absolute value of page counts for a given query, Chen *et al*., (H.-H. Chen et al., 2006) rely on the amount of co-occurrences observed in the, apparently, most reliable resources: those presented in the first positions of the results list by the search engine. For two terms *a* and *b*, they collect a fixed number of snippets provided by the search engine when querying each term. Snippets are brief windows of text extracted by a search engine around the query term in a document and provide, in a direct manner, a local context for the queried term. Snippet processing is very efficient when compared to the cost of accessing and downloading individual Web documents. Then, they count the number of occurrences of *a* in the snippets of *b f(a@b)* and vice-versa *f(b@a)*. The two values are combined in a non-linear fashion to compute their relatedness (with a function called CODC).

$$CODC(a,b) = \begin{cases} 0 & if\ f(a@b) = 0 \quad or \quad f(b@a) = 0 \\ e^{\log\left(\frac{f(b@a)}{f(a)} \times \frac{f(a@b)}{f(b)}\right)^{\alpha}} & otherwise \end{cases} \quad (28)$$

, where *f* represents the number of occurrences of the corresponding term in the top *N* snippets returned by the search engine when querying the term. The constants α=0.15 and *N*=600 were used in their experiments (H.-H. Chen et al., 2006).

This approach heavily depends on the Web search engine ranking algorithm and the fact that only a subset of snippets can be processed (*i.e.*, most search engines only provide access to the first 1000 Web resources for a given query). Therefore, there is no guarantee that the evidence needed to support the semantic assessment for a pair of terms will be contained in the top-ranked snippets. As a result, even though this method is able to provide relatively reliable results for common and related terms, it suffers from high data sparseness due to the locality of the analysis.

In a more elaborated approach, Bollegala *et al*., (Bollegala et al., 2007) also relied on snippets obtained when querying both terms, *a* and *b*, at the same time. Snippets are used as co-occurrence context, and lexical patterns (n-grams in a window from 2 to 5 words), evidencing the co-occurrence of *a* and *b*, are extracted. The most reliable patterns according to a predefined list are selected, and the number of their appearances is normalized. They create a feature vector using 200 patterns and the Web scores for *a* and *b* computed from the functions Web-Dice (26), Web-overlap (25), Web-Jaccard (24) and Web-PMI (27) stated above. The vector is created for a pre-tagged set of synonym and non-synonym word pairs and a SVM is trained accordingly. The trained SVM is then used to classify new word pairs using the same vector-based procedure. Semantic relatedness (referred with the name *SemSim*) is computed as the posterior probability *Prob(F|synonymous)* that the obtained feature vector *F* belongs to the synonymous-word class (29).

$$SemSim(a,b) = Prob(F \mid synonymous) \tag{29}$$

In Bollegala *et al*., (Bollegala et al., 2009) the same authors modified their measure by: *1)* introducing an algorithm to select the most reliable lexical patterns according to a set of semantically related words which are used as training data, and *2)* clustering semantically related patterns into groups in order to overcome data sparseness of a fine-grained pattern list and reduce the number of training parameters. As a result, two words are represented by a feature vector defined over the clusters of patterns. Semantic relatedness is computed as the Mahalanobis distance between the points of the feature vectors.

It is important to note that Bollegala *et al*.,'s supervised measures cannot be compared in the same terms as the simpler scores presented above, as authors rely on pre-tagged data and trained classifiers. This introduces many limitations such as the fact that manually tagged training data should be available and that this data should be general and big enough to avoid the risk that the classifier could be overfitted by them. As a result, the same problems noted for the IC corpus-based measures can be noted in this case.

In general, the main advantage of co-occurrence-based approaches is that, relying uniquely on the Web, they do not need any knowledge source to support the assessment. Thanks to the Web coverage of almost any written word, they can be applied to terms that are not typically considered in ontologies such as named entities. However, their unsupervised nature and their reliance on search engine page counts introduce several drawbacks. On one hand, word co-occurrence estimated by page-counts omits the semantic dimension of the co-occurrence. Words may co-occur because they are taxonomically related, but also because they are

antonyms or by pure chance. So, page counts (without considering the relative positions of words in the document) give a rough estimation of statistical dependency. On the other hand, page counts deal with words rather than concepts (on the contrary to ontological features). Due to the ambiguity of language and the omission of word context analysis during the relatedness assessment, polysemy and synonymy may negatively affect the estimation of concept probability by means of word appearance frequency. Polysemic words associated to a concept cause that their page counts contain a combination of all their senses. Moreover, the presence of synonyms for a given concept causes that word page counts underestimate the real concept probability. Finally, as stated above, page counts may not be necessarily equal to word frequency because the queried word might appear several times on a Web resource. Due to these reasons, some authors have questioned the usefulness of page counts alone as a measure of relatedness (Bollegala et al., 2007).

In (Bollegala et al., 2007; Lemaire and Denhière, 2006) the effectiveness of relying on first order co-occurrences as a measure of relatedness is also questioned. Studies on large corpora gave examples of strongly associated words that never co-occur (Lund and Burgess, 1996). This situation is caused, in many cases, by the fact that both words tend to co-occur with a third one. Psycholinguistics researchers have shown that, in those cases, the association between two words is done by means of a third word (Livesay and Burgess, 1998). This is called a *second-order co-occurrence* (Lemaire and Denhière, 2006), which is precisely the principle of the approaches reviewed in the following section.

4.2.2 Second order co-occurrence

Second order co-occurrence measures are based on the principle that two words are similar to the extent that their contexts are similar. The definition of context may vary from one measure to another and might be considered a small or large window around a word occurrence or an entire document.

A classical approach based on this principle is Latent Semantic Analysis (LSA) (Deerwester et al., 2000). It consists on compiling a term context matrix containing the occurrences of each word in each context. A Singular Value Decompositions (SVD) process is performed to enhance the differences between reliable and unreliable extractions. Considering word context as vectors, the final distance between words is computed as the cosine of the angle between them.

Using the Web as corpus, Sahami and Heilman (Sahami and Heilman, 2006) computed the likeness between two terms by means of snippets returned when querying those terms in a search engine. Authors process each snippet and represent it as a TF-IDF weighted word vector. The centroid of the set of vectors obtained by querying each term is defined, and the relatedness between two terms is computed as the inner product between the corresponding centroids.

Even though using the Web as a corpus and search engines as middlewares has several advantages derived from the Web's size and heterogeneity, some authors have criticized their usefulness as a support for relatedness computation. While semantic relatedness is inherently a relation on concepts, Web-based approaches measure a relation on words (Budanitsky and Hirst, 2006) . A big-enough sense-tagged corpora is

needed to obtain reliable concept distributions from word senses, much like corpus-based IC measures needed in the past. However, due to the nature of the Web, it is not feasible to have such tagged corpora, at least until the Semantic Web (Berners-Lee et al., 2001) becomes a reality. Moreover, ontology-based measures rely on pre-defined knowledge sources manually created by human experts, which one may consider to be true and unbiased. The Web, on the other hand, is not. As stated above, commercial bias, spam, noise and data sparseness are problems that may affect distributional measures when using the Web as corpora.

In order to overcome those problems, some authors preferred to apply distributional hypotheses over more reliable corpora. Concept glosses from wide thesaurus like WordNet were exploited. Glosses are brief and explanatory notes about the meaning of a particular word sense. Words appearing in a gloss are likely to be more relevant for the concept's meaning than text drawn from a generic corpus and, in consequence, may represent a more reliable context. Based on the distributional hypothesis, if two terms have similar glosses (*i.e.*, their textual descriptions overlap), they are likely to have similar meanings.

Banerjee and Pedersen (Banerjee and Pedersen, 2003) presented the Extended Gloss Overlap (EGO) measure (30), which determines the relatedness of terms as a function of the overlap of their WordNet glosses. As synset glosses in WordNet tend to be rather short, they extended the gloss by including example sentences (also provided by WordNet) and glosses of related concepts directly linked by means of a semantic relation.

$$EGO(a,b) = score(gloss(a), gloss(b)) + score(hyper(a), hyper(b)) + \\ + score(hypo(a), hypo(b)) + score(hyper(a), gloss(b)) + score(gloss(a), hyper(b)) \quad (30)$$

, where *score()* is the function that find the phrases that overlap between two glosses and returns a score as defined in (Banerjee and Pedersen, 2003); *hypo(a)* and *hyper(a)* represent respectively hyponyms and hypernyms of *a* in the given ontology.

Patwardhan and Pedersen (Patwardhan and Pedersen, 2006) also used extended WordNet glosses as corpora to retrieve co-occurrence information for term contexts, creating gloss vectors (GV). Gloss vectors are constructed considering gloss words that are not a stop word and whose occurrence is above a minimum frequency. Due to the size of WordNet and the extension of glosses (which consist on approximately 1.4 million words once low frequency and stop words are removed), vectors are defined in a space of 20,000 dimensions. The relatedness between two words is defined as the cosine of the angle between gloss vectors (31).

$$GV(a,b) = \frac{\vec{v}_a \cdot \vec{v}_b}{|\vec{v}_a| \cdot |\vec{v}_b|} \quad (31)$$

, where $\vec{v}_a$ and $\vec{v}_b$ are the context vectors corresponding to *a* and *b* respectively.

The Gloss Vector measure presents some advantages over the Extended Gloss Overlap, as the later looks for exact string overlaps as a measure of relatedness. Gloss Vector does not rely on exact matches by using vectors that capture the contextual representation of concepts.

Wan and Angryk (Wan and Angryk, 2007) identified some weaknesses of Patwardhan and Pedersen's measure and proposed and new Context Vector measure based on a similar principle. They used related synsets instead of their glosses to augment the gloss of a term. At the end, they join the term synset and synsets having direct semantic relations to the concerned term synset, together with all direct and inherited hypernyms. In order to limit the vector space, they remove senses with a frequency of appearance in a corpus lower than a threshold. Finally, the cosine of the angle between vectors is used as relatedness measure as in the previous formula (30).

As it will be discussed in the evaluation section, the use of reliable glosses instead of the Web as corpora results in a significant improvement of accuracy. However, the computational complexity is a factor that hampers those measures as the creation of context vectors in such a big dimensional space is considerable. Moreover, the quality of the words used as the dimensions of these vectors greatly influences the accuracy of the results. Big differences were observed by the authors (Patwardhan and Pedersen, 2006) when changing the frequency cut-off for scarce senses. Finally, by relying on WordNet glosses, those measures are hardly applicable to other ontologies in which glosses or textual descriptions are typically omitted (Ding et al., 2004). In fact, Pedersen *et al*., (Pedersen et al., 2007) applied the Gloss Vector measure to the biomedical domain by exploiting the SNOMED-CT repository as ontology. Due to the lack of concept glosses, they required a time-consuming process of manual compilation and processing of a large set of medical diagnoses from which to extract term descriptions. In that case, the algorithm parameters, such as the choice and size of corpora, had a very notorious influence in the results.

4.3 Evaluation

As stated in (Bollegala et al., 2009), an objective evaluation of the accuracy of a semantic similarity function is difficult because the notion of similarity is subjective. In order to enable fair comparisons, several authors have created evaluation benchmarks consisting of word pairs whose similarity was assessed by a set of humans. Rubenstein and Goodenough (Rubenstein and Goodenough, 1965) defined the first experiment in 1965 in which a group of 51 students, all native English speakers, assessed the similarity of 65 word pairs selected from ordinary English nouns on a scale from 0 (semantically unrelated) to 4 (highly synonymous). Miller and Charles (Miller and Charles, 1991) re-created the experiment in 1991 by taking a subset of 30 noun pairs whose similarity was reassessed by 38 undergraduate students. The correlation obtained with respect to Rubenstein and Goodenough's experiment was 0.97. Resnik (Resnik, 1995) replicated again the same experiment in 1995, in this case, requesting 10 computer science graduate students and post-doc researchers to assess similarity. The correlation with respect to Miller and Charles results was 0.96. Finally, Pirro (Pirró, 2009) replicated and compared the three above experiments in 2009, involving 101 human subjects, both English and non-English native speakers. He obtained an average correlation of 0.97 with respect to Rubenstein and Goodenough's experiment, and 0.95 with respect to Miller and Charles' experiment. It is interesting to see the high correlation obtained between the experiments even though being performed in a period of more than 40 years and with heterogeneous sets of people. This means that the similarity between the selected words is stable over the years, making them a reliable source for comparing measures.

Rubenstein and Goodenough's and Miller and Charles' benchmarks have become *de facto* tests to evaluate and compare the accuracy of similarity measures. The correlation values obtained against those benchmarks can be used to numerically quantify the closeness of two ratings sets (*i.e.*, the human judgments and the results of the computerized assessment). If the two rating sets are exactly the same, the correlation coefficient is 1, whereas 0 means that there is no relation. Spearman's and Pearson's correlations coefficients have been commonly used in the literature; both are equivalent if the ratings sets are ordered (which is the case). They are also invariant to linear transformations which may be performed over the results such as a change between distance and similarity by changing the sign of the value or normalizing values in a range.

We have taken the correlation values reported by related works for Rubenstein and Goodenough's and Miller and Charles' benchmarks (when available) and summarized them in Table 4. In the case in which a concrete measure depends on certain parameters (such as weights or corpora selection/processing) the best correlation value reported according to optimum parameter tuning was compiled. It is important to note that, even though some of them rely on different knowledge sources (such as tagged corpora or the Web), all the ontology-based ones use WordNet. WordNet 2 is the most common version used in related works. In cases in which the original authors used an older version (WordNet 2 was released in July 2003), we took a more recent replication of the measure evaluation performed by another author in order to enable a fair comparison. As a result, we picked up results reported by authors in papers published from 2004 to 2009.

Table 4. Correlation values. From left to right: authors, measure type, correlation with Miller and Charles' benchmark, correlation with Rubenstein and Goodenough's benchmark and reference in which those correlations where reported.

Measure	Type	M&C	R&G	Evaluated in
Rada *et al*., (*path length*)	Edge-counting	0.59	N/A	(Petrakis et al., 2006)
Wu and Palmer	Edge-counting	0.74	N/A	(Petrakis et al., 2006)
Leacock and Chodorow	Edge-counting	0.74	0.77	(Patwardhan and Pedersen, 2006)
Li *et al*.,	Edge-counting	0.82	N/A	(Petrakis et al., 2006)
Al-Mubaid and Nguyen (*sem*)	Edge-counting	N/A	0.815	(Hisham Al-Mubaid and Nguyen, 2009)
Hirst and St-Onge	Edge-counting	0.78	0.81	(Wan and Angryk, 2007)
Rodriguez and Egenhofer	Feature	0.71	N/A	(Petrakis et al., 2006)
Tversky	Feature	0.73	N/A	(Petrakis et al., 2006)
Petrakis *et al*., (*X-similarity*)	Feature	0.74	N/A	(Petrakis et al., 2006)
Resnik	IC (corpus)	0.72	0.72	(Patwardhan and Pedersen, 2006)
Lin	IC (corpus)	0.7	0.72	(Patwardhan and Pedersen, 2006)
Jiang and Conrath	IC (corpus)	0.73	0.75	(Patwardhan and Pedersen, 2006)
Resnik (IC computed as Seco *et al*.,)	IC (intrinsic)	N/A	0.829	(Zhou et al., 2008)
Lin (IC computed as Seco *et al*.,)	IC (intrinsic)	N/A	0.845	(Zhou et al., 2008)
Jiang and Conrath (IC computed as Seco *et al*.,)	IC (intrinsic)	N/A	0.823	(Zhou et al., 2008)
Resnik (IC computed as Zhou *et al*.,)	IC (intrinsic)	N/A	0.842	(Zhou et al., 2008)
Lin (IC computed as Zhou *et al*.,)	IC (intrinsic)	N/A	0.866	(Zhou et al., 2008)
Jiang and Conrath (IC computed as Zhou *et al*.,)	IC (intrinsic)	N/A	0.858	(Zhou et al., 2008)
Normalized Google Distance	1st ord. co-occ.	0.205	N/A	(Bollegala et al., 2009)
Web-Jaccard	1st ord. co-occ.	0.259	N/A	(Bollegala et al., 2007)
Web-Overlap	1st ord. co-occ.	0.382	N/A	(Bollegala et al., 2007)
Web-Dice	1st ord. co-occ.	0.267	N/A	(Bollegala et al., 2007)
Web-PMI	1st ord. co-occ.	0.548	N/A	(Bollegala et al., 2007)
Chen *et al*., (CODC)	1st ord. co-occ.	0.693	N/A	(Bollegala et al., 2007)
Bollegala *et al*., 2007 (*SemSim*)	1st ord. co-occ.	0.834	N/A	(Bollegala et al., 2007)
Bollegala *et al*., 2009	1st ord. co-occ.	0.867	N/A	(Bollegala et al., 2009)
Latent Semantic Analysis	2n ord. (Web)	0.72	N/A	(Seco et al., 2004)
Sahami and Heilman	2n ord. (Web)	0.579	N/A	(Bollegala et al., 2007)
Banerjee and Pedersen (*Extended Gloss Overlap*)	2n ord. (WordNet)	0.81	0.83	(Patwardhan and Pedersen, 2006)
Patwardhan and Pedersen (*Gloss Vector*)	2n ord. (WordNet)	0.91	0.9	(Patwardhan and Pedersen, 2006)
Wan and Angryk (*Context Vector*)	2n ord. (WordNet)	0.80	0.83	(Wan and Angryk, 2007)

4.4 Discussion

For ontology-based measures, the basic path length measure (Rada et al., 1989) presents the lowest accuracy (0.59) due to the fact that the absolute lengths of the paths between two concepts may not accurately represent their specificity. This is the case of WordNet, since concepts higher in the hierarchy are more general than those lower in the hierarchy (Pirró, 2009). As a result, other edge-counting approaches also exploiting the relative depth of the taxonomy (Wu and Palmer (Wu and Palmer, 1994), Leadcock and Chodorow (Leacock and Chodorow, 1998)) offer a higher accuracy (0.74). The correlation values obtained by Li (Y. Li et al., 2003) and Al-Mubaid and Nguyen (H. Al-Mubaid and Nguyen, 2006), which combine the length of the path with the depth of the concepts in a weighted and non-linear manner, are remarkable. However, they rely on empirical parameters whose values have been experimentally determined to optimize the accuracy for the evaluated benchmark, hampering their generality. Hirst and St-Onge (Hirst and St-Onge, 1998) present a similar behaviour, also relying on tuning parameters but, in this case, using non-taxonomic relationships that consider a more general concept of relatedness.

Feature-based methods try to overcome the limitations of path-based measures by considering different kinds of ontological features. The problem, which has been also noted for some edge-counting measures, is their dependence on the parameters introduced to weight the contribution of each feature (for the approaches of Rodriguez and Egenhofer (Rodríguez and Egenhofer, 2003) and Tversky (Tversky, 1977) approaches). Correlation values are, however, very similar to those offered by edge-counting measures (0.71-0.74) in these benchmarks. This can due to the fact that they rely on concept features, such as synsets, glosses or non-taxonomic relationships which have secondary importance in ontologies like WordNet in comparison with taxonomical features. In fact, those kinds of features are scarce in ontologies (Ding et al., 2004), which causes those approaches to be based on partially modelled knowledge. As a result, those measures, even being more complex, are not able to significantly outperform the state of the art of edge-counting measures.

For IC-based measures, we observe that the approaches relying on an intrinsic computation of IC (based on the number of concept hyponyms) clearly outperform the approaches relying on corpora (0.72 vs. 0.84, in average). This is very convenient as corpora dependency seriously hampers the applicability of classical IC measures. The difference between both ways of computing IC is caused by two factors. Firstly, the data sparseness problem that appears when relying on tagged corpora (which would be necessary small due to manual tagging) to obtain accurate concept appearance frequencies. Secondly, the fact that WordNet's taxonomy is detailed and fine-grained, which enables an accurate estimation of a term's generality as a function of its number of hyponyms. With regard to the performance of each measure, Lin's (Lin, 1998) tends to improve Resnik's (Resnik, 1995) one when IC is computed intrinsically, as the former is able to differentiate terms with identical LCS but different taxonomical depths. With regard to the way in which the intrinsic IC

is computed, more complex approaches also exploiting relative depth and relying on weighting parameters (Zhou *et al*., (Zhou et al., 2008)) offer the highest accuracy (0.86).

With regard to distributional approaches, unsupervised approaches relying on direct term co-occurrences computed from Web page counts (Web-Jaccard, Web-Overlap, Web-PMI, Web-Dice and NGD) offer a limited performance (between 0.2 and 0.54). As stated in section 2.2.1, uncontextual Web page-counts are not accurate enough to estimate reliable term resemblance due to ambiguity and noise of word Web occurrences. On the contrary, Chen *et al*., (H.-H. Chen et al., 2006), and Bollegala *et al*., works (Bollegala et al., 2007, 2009) exploit snippets as contexts in which terms co-occur. In these experiments, we can see that this approach produces a less ambiguous estimation of term co-occurrence (due to their likeness) and better accuracy (0.69 for Chen *et al*.,'s approach). Bollegala *et al*., works offer a noticeably high accuracy (0.83-0.86) as they rely on a supervised classifier (trained SVM) and lexical patterns to distinguish highly similar co-occurrent words (such as synonyms) from less related ones. Even though those methods can be applied to terms that are not typically covered by ontologies (such as named entities), their dependency on manually tagged data and trained classifiers compromise their applicability.

Distributional approaches based on second order co-occurrences computed from the Web (such as LSA) improve the results of unsupervised first order approaches (0.72 vs. 0.54). Second order co-occurrences are able to capture non-directly co-occurrent words (such as synonyms) that, even though being highly related, typically co-occur by means of a third word. When a highly reliable and structured corpus such as WordNet glosses is used instead of the more general and noisy Web, the accuracy is significantly improved. In this manner, gloss vector and gloss overlap-based approaches (Banerjee and Pedersen, 2003; Patwardhan and Pedersen, 2006; Wan and Angryk, 2007) are able to obtain correlation values among 0.8 and 0.91 in these tests. In fact, the Gloss Vector approach reported the highest correlation values ever achieved for the evaluated benchmarks (0.91 and 0.9). It is worth to noting that the Context Vector measure (Wan and Angryk, 2007), even aiming to overcome some of the theoretical limitations observed by the authors for the Gloss Vector measure, obtained a lower correlation (0.91 vs. 0.8). However, the Gloss Vector accuracy heavily depends on the way in which contexts are built. Authors (Patwardhan and Pedersen, 2006) reported a high variability on the results according to the filtering policy (*i.e.*, stop words removal and TF-IDF-based cut-offs) applied to words extracted from concept glosses. As a result, the maximum correlation value is obtained under a carefully tuned setting. The accuracy lowered down to 0.7 when TF-IDF cut-offs were modified in the authors' experiments. Another limitation is caused by their reliance on concept glosses. When this information is not directly available (which is the usual case in ontologies), word vectors are more difficult to build, requiring the compilation and processing of reliable corpora. The same authors (Pedersen et al., 2007) discussed the difficulties and dependency on corpora and parameter tuning of their measure when applied to the domain of Biomedicine. These dependencies limit the applicability of those measures in concrete domains.

Summarizing, intrinsinc IC-based measures provide a high accuracy without any dependency on data availability, data pre-processing or tuning parameters for a concrete scenario. As they only rely on the most commonly available ontological feature, they ensure their generality as a domain-independent proposal. At the same time, they retain the low computational complexity and lack of constraints of edge-counting measures as they only require retrieving, comparing and counting ontological hyponyms. This ensures its scalability when it must be used in engineering or data mining applications, which may require dealing with large sets of terms (Armengol, 2009; Batet et al., 2008).

As any other ontology-based measure, the final accuracy will depend on the detail, completeness and coherency of the taxonomical knowledge. Moreover, most of the improvements achieved by these approaches are derived from the fact that the similarity is estimated from the total set of subsumer concepts considering the different taxonomical hierarchies. If the input ontology offers little taxonomical detail, the accuracy improvements of these approaches with respect to the measures based on the minimum path are likely to be less noticeable. Fortunately, large and broad ontologies are being developed, like WordNet as a general purpose description of concepts, SNOMED-CT or MeSH in the medical context or OntoCAPE (Morbach et al., 2007) for the engineering domain.

5 Semantic similarity measures into clustering algorithms

As it has been seen in section 2, both the partitional and hierachical approaches to clustering have a critical component: the way of measuring the distance or dissimilarity between a pair of individuals. The distance between individuals (or centroids) is a clue to decide which individuals belong to the same cluster. In fact, it is in the core of the goal of the clustering, which is to find a set of clusters with similar individuals.

In this project, a data matrix with different types of values will be considered. The DAMASK project will include numerical, nominal (i.e. non-ordered categorical values) and semantic features. Semantic features are an extension of categorical features, which have a non fixed and large set of possible values, without any order or scale of measurement defined between terms.

Traditionally, the comparison between two values in categorical variables is done simply based on the equality/inequality of the words, due to the lack of proper methods for representing the meaning of the terms. Some widely used distance measures for categorical values are the Chi-Squared and the Hamming distance (Esposito et al., 2000). However, as it has been explained in section 4, nowadays there are many ways to estimate the similarity between terms from a semantic point of view. In the so-called semantic features, each of the values in the data matrix corresponds to a concept, thus, reasoning at a conceptual level should be done in order to estimate the similarity between objects during the clustering process.

When heterogeneous types of values must be taken into consideration in a joint way, two main approaches can be used:

1. The transformation of the values into a common domain (e.g. discretization of numerical variables, or mapping the data into a new space using projection algorithms (Anderberg, 1973; Anil K. Jain and Dubes, 1988)).
2. The use of compatibility measures that combine different expressions according to the type of each of the variables (Anderberg, 1973; Gibert and Cortés, 1997; Gowda and Diday, 1991; Ichino and Yaguchi, 1994).

This second approach allows the analysis of the different values maintaining the original scales, without making any transformation, having three main advantages: (1) data are analyzed in its original nature, (2) there is no a priori loss of information produced by previous transformations (i.e. discretization of numerical variables) and (3) it avoids taking previous arbitrary decisions that could bias results.

In order to take advantage of the potential of semantic similarity measures, a compatibility measure is needed to combine the contribution of numerical, nominal and semantic features into a global function. After the definition of this compatibility operation, any of the semantic similarity functions could be used to deal with the comparison of semantic values (i.e. terms corresponding to concepts). Some preliminary work in such a compatibility measure has already been proposed by the research team of this project in (Batet et al., 2008).

6 References

Al-Mubaid, H., Nguyen, H.A., 2006. A cluster-based approach for semantic similarity in the biomedical domain, in: Proc. of 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2006 IEEE Computer Society, New York, USA, pp. 2713–2717.

Al-Mubaid, H., Nguyen, H.A., 2009. Measuring Semantic Similarity between Biomedical Concepts within Multiple Ontologies. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39(4), 389-398.

Allampalli-Nagaraj, G., Bichindaritz, I., 2009. Automatic semantic indexing of medical images using a web ontology language for case-based image retrieval. Engineering Applications of Artificial Intelligence 22(1), 18-25.

Anderberg, M.R., 1973. Cluster analysis for applications. Academic Press Inc, New York.

Armengol, E., 2009. Using explanations for determining carcinogenecity in chemical compounds. Engineering Applications of Artificial Intelligence 22(1), 10-17.

Baeza-Yates, R.A., 1992. Introduction to data structures and algorithms related to information retrieval, in: W.B. Frakes , R. Baeza-Yates (Eds.), Information Retrieval: Data Structures and Algorithms. Prentice-Hall, Inc., Upper Saddle River, NJ, pp. 13–27.

Ball, G.H., Hall, D.J., 1965. ISODATA, a novel method of data analysis and classification.

Banerjee, S., Pedersen, T., 2003. Extended Gloss Overlaps as a Measure of Semantic Relatedness, in: Proc. of 18th International Joint Conference on Artificial Intelligence, IJCAI 2003. Morgan Kaufmann, Acapulco, Mexico, pp. 805-810.

Basu, S., Davidson, I., Wagstaff, K.L., 2008. Constrained Clustering: Advances in Algorithms, Theory, and Applications, in: V. Kumar (Ed.), Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC.

Batet, M., Isern, D., Valls, A., 2010. Internal project report - Task 1.2: Data Types. Data-Mining Algorithms with Semantic Knowledge (TIN2009-11005). ITAKA – Intelligent Technologies for Advanced Knowledge Acquisition (URV), Tarragona.

Batet, M., Valls, A., Gibert, K., 2008. Improving classical clustering with ontologies, in: Proc. of 4th World Conference of the IASC and 6th Conference of the Asian Regional Section of the IASC on Computational Statistics & Data Analysis, IASC 2008. International Association for Statistical Computing, Yokohama, Japan, pp. 137-146.

Beliakov, G., Bustince, H., Fernández, J., 2010. On the median and its extensions, Computational Intelligence for Knowledge-based Systems Design, LNAI 6178. Springer, pp. 435-444.

Berners-Lee, T., Hendler, J., Lassila, O., 2001. The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American 284(5), 34-43.

Bichindaritz, I., Akkineni, S., 2006. Concept mining for indexing medical literature. Engineering Applications of Artificial Intelligence 19(4), 411-417.

Blank, A., 2003. Words and Concepts in Time: Towards Diachronic Cognitive Onomasiology, in: R. Eckardt, K. von Heusinger , C. Schwarze (Eds.), Words and Concepts in Time: towards Diachronic Cognitive Onomasiology. Mouton de Gruyter, Berlin, Germany, pp. 37-66.

Bollegala, D., Matsuo, Y., Ishizuka, M., 2007. Measuring Semantic Similarity between Words Using Web Search Engines, in: Proc. of 16th international conference on World Wide Web, WWW 2007. ACM, Banff, Alberta, Canada pp. 757-766.

Bollegala, D., Matsuo, Y., Ishizuka, M., 2009. A Relational Model of Semantic Similarity between Words using Automatically Extracted Lexical Pattern Clusters from the Web, in: Proc. of Conference on Empirical Methods in Natural Language Processing, EMNLP 2009. ACL and AFNLP, Singapore, Republic of Singapore, pp. 803–812.

Budanitsky, A., Hirst, G., 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, in: Proc. of Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics. Pittsburgh, USA, pp. 10-15.

Budanitsky, A., Hirst, G., 2006. Evaluating wordnet-based measures of semantic distance. Computational Linguistics 32(1), 13-47.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S., 2003. Model-Based Clustering and Visualization of Navigation Patterns on a Web Site. Data Mining and Knowledge Discovery 7(4), 399-424.

Calinski, R.B., Harabasz, J., 1974. A dendrite method for cluster analysis. Communications in Statistics 3, 1–27.

Carpineto, C., Osinski, S., Romano, G., Weiss, D., 2009. A survey of Web clustering engines. ACM Computing Surveys 41(3), 1-38.

Cilibrasi, R.L., Vitányi, P.M.B., 2006. The Google Similarity Distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370-383.

Cimiano, P., Handschuh, S., Staab, S., 2004. Towards the self-annotating web, in: Proc. of 13th international conference on World Wide Web, WWW 2004. ACM, New York, USA, pp. 462 - 471.

Corchado, J.M., Fyfe, C., 2000. A comparison of kernel methods for instantiating case based reasoning systems. Computing and Information Systems 7, 29-42.

Curran, J.R., 2002. Ensemble Methods for Automatic Thesaurus Extraction, in: Proc. of Conference on Empirical Methods in Natural Language Processing, EMNLP 2002. Association for Computational Linguistics, Philadelphia, PA, USA, pp. 222–229.

Chavent, M., Lechevallier, Y., Briant, O., 2007. DIVCLUS-T: A monothetic divisive hierarchical clustering method. Computational Statistics & Data Analysis 52(2), 687-701.

Chen, H.-H., Lin, M.-S., Wei, Y.-C., 2006. Novel association measures using web search with double checking, in: Proc. of 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, COLING-ACL 2006. ACL, Sydney, Australia, pp. 1009-1016.

Chen, J.Y., Lonardi, S., 2009. Biological Data Mining, in: V. Kumar (Ed.), Mining and Knowledge Discovery Series. Chapman & Hall/CRC.

Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L., 2009. Similarity-based classification: Concepts and Algorithms. Journal of Machine Learning Research 10(Mar), 747-776.

Day, W.H.E., 1992. Complexity theory: An introduction for practitioners of classification, in: P. Arabie , L. Hubert (Eds.), Clustering and Classification. World Scientific Publishing Co., Inc, River Edge, NJ.

Deerwester, S., Dumais, S.T., Harshman, R., 2000. Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391-407.

Dillon, W.R., Goldstein, M., 1984. Multivariate Analysis: Methods and Applications. Wiley.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J., 2004. Swoogle: A Search and Metadata Engine for the Semantic Web, in: Proc. of thirteenth ACM international conference on Information and knowledge management, CIKM 2004. ACM Press, Washington, D.C., USA, pp. 652-659.

Domingo-Ferrer, J., Torra, V., 2003. Median based aggregation operators for prototype construction in ordinal scales. International Journal of Intelligent Systems 18(6), 633-655.

Esposito, F., Malerba, D., Tamma, V., Bock, H.H., 2000. Similarity and dissimilarity, in: H.H.B.a.E. Diday (Ed.), Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data. Springer, New York, NY, pp. 139-152.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, Portland, Oregon, USA, pp. 226–231.

Everitt, B.S., Landau, S., Leese, M., 2001. Cluster Analysis. Arnold, London.

Fan, B., 2009. A hybrid spatial data clustering method for site selection: The data driven approach of GIS mining. Expert Systems with Applications 36(2), 3923-3936.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., 1996. From Data Mining to Knowledge Discovery in Databases. AI Magazine 17(3), 37-54.

Fellbaum, C., 1998. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts.

Fisher, D., 1987. Knowledge acquisition via incremental conceptual clustering. Maching Learning 2, 139–172.

Forgy, E., 1965. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics 21, 768–780.

Gibert, K., Cortés, U., 1997. Weighing quantitative and qualitative variables in clustering methods. Mathware and Soft Computing 4(3), 251-266.

Gibert, K., Silva, G.R., Rodríguez-Roda, I., 2010. Knowledge discovery with clustering based on rules by states: A water treatment application. Environmental Modelling and Software 25(6), 712-723.

Godo, L., Torra, V., 2000. On aggregation operators for ordinal qualitative information. IEEE Transactions on Fuzzy Systems 8(2), 143-154.

Goldstone, R.L., 1994. Similarity, interactive activation, and mapping. Journal of Experimental Psychology: Learning, Memory, and Cognition 20(1), 3-28.

Gowda, K.C., Diday, E., 1991. Symbolic clustering using a new similarity measure. IEEE Transactions on Systems, Man and Cybernetics 22.

Gower, J.C., 1967. A comparison of some methods of cluster analysis. Biometrics 23, 623-628.

Guarino, N., 1998. Formal Ontology in Information Systems, in: Proc. of 1st International Conference on Formal Ontology in Information Systems, FOIS 1998. IOS Press, Trento, Italy, pp. 3-15.

Guha, S., Rastogi, R., Shim, K., 2000. ROCK: A robust clustering algorithm for categorical attributes. Information Systems 25(5), 345–366.

Guha, S., Rastogi, R., Shim, K., 2001. CURE: An efficient clustering algorithm for large databases. Information Systems 26(1), 35-58.

Gupata, S., Rao, K., Bhatnagar, V., 1999. K-means clustering algorithm for categorical attributes, in: Proc. of 1st Int. Conf. Data Warehousing and Knowledge Discovery (DaWaK'99). Florence, Italy, pp. 203–208.

Hamasuna, Y., Endo, Y., Miyamoto, S., 2010. On tolerant fuzzy c-means clustering and tolerant possibilistic clustering. Soft Computing 14(5), 487-494.

Han, J., Kamber, M., Tung, A., 2001. Spatial Clustering Methods in Data Mining: A Survey, in: H.J. Miller , J. Han (Eds.), Geographic Data Mining and Knowledge Discovery. Taylor & Francis, London, pp. 201-231.

Hansen, P., Jaumard, B., 1997. Cluster analysis and mathematical programming. Mathematical Programming 79(1-3), 191–215.

Hirst, G., St-Onge, D., 1998. Lexical chains as representations of context for the detection and correction of malapropisms, in: C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database. MIT Press, pp. 305–332.

Hliaoutakis, A., 2005. Semantic Similarity Measures in the MESH Ontology and their Application to Information Retrieval on Medline. Diploma Thesis. Technical Univ. of Crete (TUC), Dept. of Electronic and Computer Engineering, Crete, Greece.

Huang, H., Cheng, Y., Zhao, R., 2008. A Semi-supervised Clustering Algorithm Based on Must-Link Set, in: Proc. of 4th International Conference on Advanced Data Mining and Applications, ADMA 2008. LNAI 5139. Springer, Chengdu, China, pp. 492-499.

Huang, Z., 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery 2, 283–304.

Ichino, M., Yaguchi, H., 1994. Generalized Minkowski Metrics for Mixed feature-type data analysis. IEEE Transaction on Systems, Man and Cybernetics 22(2), 146–153.

Jain, A.K., Dubes, R.C., 1988. Algorithms for clustering data. Prentice-Hall, Michigan, USA.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Computing Surveys 31(3), 264-323.

Jiang, J.J., Conrath, D.W., 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, in: Proc. of International Conference on Research in Computational Linguistics, ROCLING X. Taiwan, pp. 19-33.

Karypis, G., Han, E., Kumar, V., 1999. Chameleon: Hierarchical clustering using dynamic modeling. IEEE Computer 32(8), 68–75.

Kaufman, L., Rousseeuw, P., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley, New York.

Keller, F., Lapata, M., 2003. Using the web to obtain frequencies for unseen bigrams. Computational Linguistics 29(3), 459-484.

Kimura, M., Saito, K., Nakano, R., Motoda, H., 2010. Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20(1), 70-97.

Krishna, K., Murty, M., 1999. Genetic K-means algorithm. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 29(3), 433–439.

Lance, G.N., Williams, W.T., 1967. A general theory of classificatory sorting strategies: II. Clustering algorithms. Computer Journal 10, 271-277.

Landauer, T., Dumais, S., 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge. Psychological Review 104, 211-240.

Leacock, C., Chodorow, M., 1998. Combining local context and WordNet similarity for word sense identification, WordNet: An electronic lexical database. MIT Press, pp. 265-283.

Lemaire, B., Denhière, G., 2006. Effects of High-Order Co-occurrences on Word Semantic Similarities. Current Psychology Letters - Behaviour, Brain and Cognition 18(1), 1.

Li, C., Biswas, G., 1999. Temporal pattern generation using hidden Markov model based unsupervised classification, in: D. Hand, K. Kok , M. Berthold (Eds.), Advances in Intelligent Data Analysis: Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis. LNCS 1642. Springer Berlin, pp. 245-256.

Li, C., Biswas, G., 2002. Unsupervised learning with mixed numeric and nominal data. IEEE Transactions on Knowledge and Data Engineering 14(4), 673–690.

Li, Y., Bandar, Z., McLean, D., 2003. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. IEEE Transactions on Knowledge and Data Engineering 15(4), 871-882.

Lin, D., 1998. An Information-Theoretic Definition of Similarity, in: Proc. of Fifteenth International Conference on Machine Learning, ICML 1998. Morgan Kaufmann, Madison, Wisconsin, USA, pp. 296-304.

Livesay, K., Burgess, C., 1998. Mediated priming in high-dimensional semantic space: No effect of direct semantic relationships or co-occurrence. Brain and Cognition 37, 102-105.

Lund, K., Burgess, C., 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research, Methods, Instruments and Computers 28(2), 203-208.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, in: Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297.

Mika, P., 2007. Ontologies are us: A unified model of social networks and semantics. Web Semantics: Science, Services and Agents on the World Wide Web 5(1), 5-15.

Miller, G.A., Charles, W.G., 1991. Contextual correlates of semantic similarity. Language and Cognitive Processes 6(1), 1-28.

Mirkin, B., 2005. Clustering for data mining: a data recovery approach. Chapman & Hall/CRC, London.

Mollineda, R., Vidal, E., 2000. A relative approach to hierarchical clustering, in: M. Torres , A. Sanfeliu (Eds.), Pattern Recognition and Applications, Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, The Netherlands.

Monreale, A., Andrienko, G., Andrienko, N., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S., 2010. Movement Data Anonymity through Generalization. Transactions on Data Privacy 3(2), 91-121.

Morbach, J., Yang, A., Marquardt, W., 2007. OntoCAPE—A large-scale ontology for chemical process engineering. Engineering Applications of Artificial Intelligence 20(2), 147-161.

Mori, J., Ishizuka, M., Matsuo, Y., 2007. Extracting keyphrases to represent relations in social networks from web, in: Proc. of 20th International Joint Conference on Artificial Intelligence, IJCAI 2007. AAAI Press, Hyderabad, India, pp. 2820-2825.

Murty, M.N., Krishna, G., 1980. A computationally efficient technique for data clustering. Pattern Recogn. 12, 153–158.

Nagy, G., 1968. Proceedings of the IEEE. State of the art in pattern recognition 56(5), 836–862.

Naphade, M.R., Huang, T.S., 2001. Semantic filtering of video content, in: Proc. of Storage and Retrieval for Multimedia Databases. SPIE, San Jose, CA, USA, pp. 270--279.

Patwardhan, S., Banerjee, S., Pedersen, T., 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation, in: Proc. of 4th International Conference on Computational Linguistics and Intelligent Text Processing and Computational Linguistics, CICLing 2003. Springer Berlin / Heidelberg, Mexico City, Mexico, pp. 241-257.

Patwardhan, S., Pedersen, T., 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts, in: Proc. of EACL 2006 Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together. Trento, Italy, pp. 1-8.

Pedersen, T., Pakhomov, S., Patwardhan, S., Chute, C., 2007. Measures of semantic similarity and relatedness in the biomedical domain. Journal of Biomedical Informatics 40(3), 288-299.

Pelleg, D., Moore, A., 2000. X-means: Extending K-means with efficient estimation of the number of clusters, in: Proc. of 17th International Conference on Machine Learning (ICML 2000). Morgan Kaufmann, Stanford, CA, USA, pp. 727–734.

Petrakis, E.G.M., Varelas, G., Hliaoutakis, A., Raftopoulou, P., 2006. X-Similarity:Computing Semantic Similarity between Concepts from Different Ontologies. Journal of Digital Information Management 4, 233-237.

Pirró, G., 2009. A semantic similarity metric combining features and intrinsic information content. Data & Knowledge Engineering 68(11), 1289-1308

Pirró, G., Seco, N., 2008. Design, Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content, in: Proc. of OTM 2008 Confederated International Conferences CoopIS, DOA, GADA, IS, and ODBASE 2008. Springer Berlin / Heidelberg, Monterrey, Mexico, pp. 1271-1288.

Rada, R., Mili, H., Bichnell, E., Blettner, M., 1989. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics 9(1), 17-30.

Resnik, P., 1995. Using Information Content to Evalutate Semantic Similarity in a Taxonomy, in: Proc. of 14th International Joint Conference on Artificial Intelligence, IJCAI 1995. Morgan Kaufmann Publishers Inc. , Montreal, Quebec, Canada, pp. 448-453.

Rodríguez, M.A., Egenhofer, M.J., 2003. Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering 15(2), 442–456.

Romdhane, L.B., Shili, H., Ayeb, B., 2010. Mining microarray gene expression data with unsupervised possibilistic clustering and proximity graphs. Applied Intelligence 33(2), 220-231.

Rubenstein, H., Goodenough, J., 1965. Contextual correlates of synonymy. Communications of the ACM 8(10), 627-633.

Sahami, M., Heilman, T.D., 2006. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets, in: Proc. of 15th International World Wide Web Conference, WWW 2006. ACM Press, Edinburgh, Scotland pp. 377 - 386

Sánchez, D., Moreno, A., 2007. Bringing taxonomic structure to large digital libraries. International Journal of Metadata, Semantics and Ontologies 2(2), 112-122.

Sánchez, D., Moreno, A., 2008. Learning non-taxonomic relationships from web documents for domain ontology construction. Data & Knowledge Engineering 63(3), 600-623.

Sato, M., Sato, Y., Jain, L., 1997. Fuzzy Clustering Models and Applications Vol. 9.

Schallehn, E., Sattler, K.-U., Saake, G., 2004. Efficient similarity-based operations for data integration. Data & Knowledge Engineering 48(3), 361-387.

Seco, N., Veale, T., Hayes, J., 2004. An Intrinsic Information Content Metric for Semantic Similarity in WordNet, in: Proc. of 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004. IOS Press, Valencia, Spain, pp. 1089-1090.

Sneath, P., 1957. The application of computers to taxonomy. Journal of General Microbiology 17, 201–226.

Sneath, P.H.A., Sokal, R.R., 1973. Numerical Taxonomy. Freeman, London,UK.

Sokal, R.R., Michener, C.D., 1958. A statistical method for evaluating systematic relationships. University Kansas Scientific Bulletin 38, 1409-1438.

Sorensen, T., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyzes of the vegetation on Danish commons. Biologiske Skrifter 5, 1–34.

Stevenson, M., Greenwood, M.A., 2005. A semantic approach to IE pattern induction, in: Proc. of 43rd Annual Meeting on Association for Computational Linguistics, COLING-ACL 2005. Association for Computational Linguistics, Ann Arbor, Michigan, USA, pp. 379–386.

Tirozzi, B., Bianchi, D., Ferraro, E., 2007. Introduction to computational neurobiology and clustering, Series on Advances in Mathematics for Applied Sciences. World Scientific Publishing, Singapore.

Turney, P.D., 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL, in: Proc. of 12th European Conference on Machine Learning, ECML 2001. Springer-Verlag, Freiburg, Germany, pp. 491-502.

Tversky, A., 1977. Features of Similarity. Psychological Review 84(4), 327-352.

Valls, A., Batet, M., López, E.M., 2009. Using expert's rules as background knowledge in the ClusDM methodology. European Journal of Operational Research 195, 864–875.

Waltinger, U., Cramer, I., Wandmacher, T., 2009. From Social Networks To Distributional Properties: A Comparative Study On Computing Semantic Relatedness, in: Proc. of Thirty-First Annual meeting of the

Cognitive Science Society, CogSci 2009. Cognitive Science Society, Amsterdam, Netherlands, pp. 3016-3021.

Wan, S., Angryk, R.A., 2007. Measuring Semantic Similarity Using WordNet-based Context Vectors, in: Proc. of IEEE International Conference on Systems, Man and Cybernetics, SMC 2007. IEEE Computer Society, Montreal, Quebec, Canada, pp. 908 - 913.

Ward, J.H., 1963. Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58, 236-244.

Wu, Z., Palmer, M., 1994. Verb semantics and lexical selection, in: Proc. of 32nd annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Las Cruces, New Mexico, pp. 133 -138.

Xu, R., Wunsch, D., 2005. Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645-678.

Zadeh, L.A., 1965. Fuzzy Sets. Information and Control 8(3), 338-353.

Zahn, C.T., 1971. Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Transactions on Computers, 68–86.

Zhang, T., Ramakrishnan, R., Linvy, M., 1997. BIRCH: An efficient data clustering method for very large data sets. Data Mining and Knowledge Discovery 1(12), 141–182.

Zhou, Z., Wang, Y., Gu, J., 2008. A New Model of Information Content for Semantic Similarity in WordNet, in: Proc. of Second International Conference on Future Generation Communication and Networking Symposia, FGCNS 2008. IEEE Computer Society, Sanya, Hainan Island, China, pp. 85-89.