Informative Polythetic Hierarchical Ephemeral Clustering

Gaël Dias*†, Guillaume Cleuziou‡ and David Machado*
*HULTIG
University of Beira Interior, Covilhã, Portugal
Email: {ddg, david}@hultig.di.ubi.pt
†DLU - GREYC
University of Caen Basse-Normandie, Caen, France
‡LIFO
University of Orléans, Orléans, France
Email: guillaume.cleuziou@univ-orleans.fr

***Abstract*—Ephemeral clustering has been studied for more than a decade, although with low user acceptance. According to us, this situation is mainly due to (1) an excessive number of generated clusters, which makes browsing difficult and (2) low quality labeling, which introduces imprecision within the search process. In this paper, our motivation is twofold. First, we propose to reduce the number of clusters of Web page results, but keeping all different query meanings. For that purpose, we propose a new polythetic methodology based on an informative similarity measure, the InfoSimba, and a new hierarchical clustering algorithm, the $HISGK$-means. Second, a theoretical background is proposed to define meaningful cluster labels embedded in the definition of the $HISGK$-means algorithm, which may elect as best label, words outside the given cluster. To confirm our intuitions, we propose a new evaluation framework, which shows that we are able to extract most of the important query meanings but generating much less clusters than state-of-the-art systems.**

***Keywords*-Hierarchical Ephemeral Clustering; Polythetic Web Snippet Representation; Informative Similarity Measure; Automatic Cluster and Label Evaluation**

I. Introduction

With so much information available on the Web, in particular with the explosion of weblogs and social networks, looking for relevant information on the internet has become more and more difficult for the last years. Indeed, traditional Web search engines still return lists of ranked documents represented by their titles and corresponding Web snippets, from which users have to go through extensively to find the documents that most satisfy their needs. To avoid what has turned to be a tedious task, some search engines such as Yippy (http://www.yippy.com), Carrot (http://carrot2.org), iBoogie (http://www.iboogie.com), SnakeT (http://snaket.di.unipi.it) or Vipaccess (http://hultig.di.ubi.pt/vipaccess) propose to help users in their process of seeking for information, by digesting Web page results through the dynamic generation of taxonomic structures. Known as post-retrieval document browsing or ephemeral clustering [1], this process constructs flat or hierarchical taxonomies from sets of Web page results, which evidence a short life span and are usually used for interactive browsing purposes.

Although, ephemeral clustering has been studied for more than a decade, it has received low user acceptance. According to us, there are two main reasons for this situation. First, state-of-the-art systems tend to generate an excessive number of clusters. As a consequence, browsing through a high number of clusters is mostly similar to searching through a high number of Web pages. Second, improved user interfaces can only be achieved through high quality cluster labeling. In the optimal case, the labels of the clusters should clearly evidence their overall contents. However, very little has been proposed to overcome this situation.

In this paper, we want to go further in the analysis of Web page results by generating clusters, which should embody exactly one sense or one sub-topic of the query. As a consequence, all sub-topics (resp. sub-meanings i.e. hyponyms) of a given query topic (resp. meaning) should be structured as sub-clusters of the main topic (resp. meaning). For example, in the optimal case, any system should mainly propose two clusters for the query *interpol*, i.e. the cluster for the sense of *police organization* and the sense of *music group*. While the other implemented state-of-the-art systems [2] [3] [4] [5] and Yippy propose a large number of mixed unorganized flat or hierarchical clusters, our $HISGK$-means algorithm clearly identifies two main clusters ($\langle$*icpo, access, police organization*$\rangle$ and $\langle$*news, music, interpol music*$\rangle$). This situation is illustrated in Figures 1, 2 and 3.

In order to achieve such results, we propose to evaluate similarity between Web snippets based on an informative similarity measure. While existing methodologies evaluate the similarity between Web snippets based on the exact match of constituents (i.e. relevant sequences of strings), we propose that two Web snippets are highly related if both share highly related (eventually different) constituents. As a consequence, similarity is not any more based on the exact match of constituents but on related words. For that purpose, we propose to use the InfoSimba similarity measure [6] to evaluate the similarity between Web snippets.

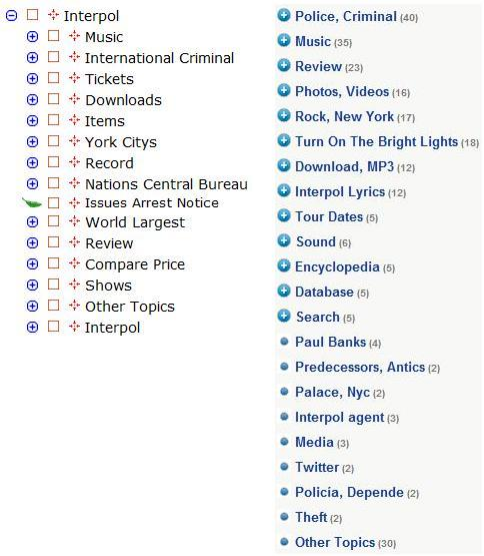

Figure 1. Clusters of SnakeT [4] (Left) and Yippy (Right) [retrieved on $22^{nd}January, 2011$].

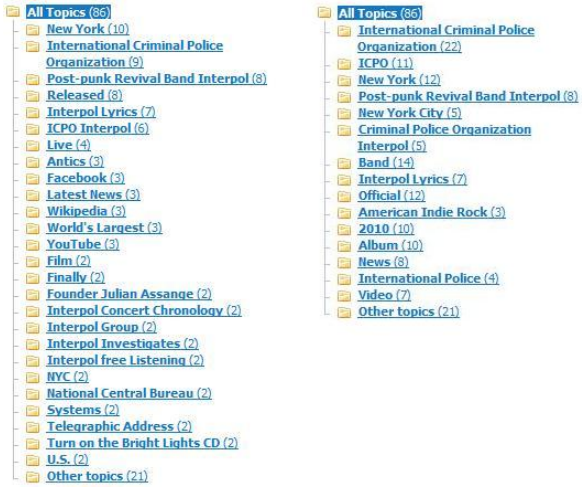

Figure 2. Clusters of Lingo [3] (Left) and STC (Right) [2] [retrieved on $22^{nd}January, 2011$ from http://carrot2.org].

So, after building a Web snippet $\times$ Web snippet informative similarity matrix, we apply a new hierarchical divisive hard clustering algorithm called the hierarchical InfoSimba-based global K-means ($HISGK$-means) for which we provide a well-founded mathematical background, which guarantees optimal clustering. In particular, the hierarchical process is a top-down approach, which recursively splits a set of Web snippets based on a variant of the global K-means algorithm (GK-means) [7] combined with the InfoSimba informative similarity measure, which we call the InfoSimba-based global K-means ($ISGK$-means). Afterwards, for each generated cluster we extract a small set of

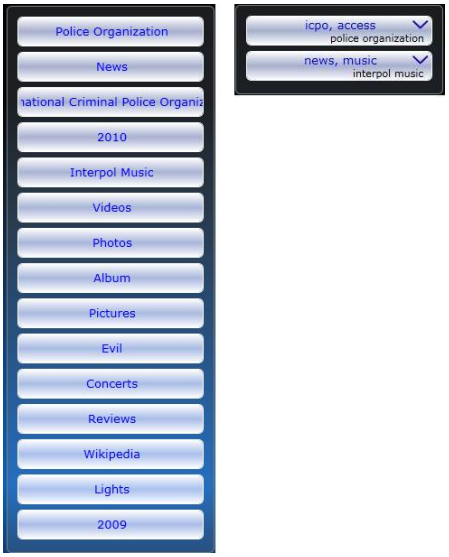

Figure 3. Clusters of CBL [5] (Left) and $HISGK$-means (Right) [retrieved on $22^{nd}January, 2011$].

representative words (i.e. the cluster labels) based on a new selection strategy with the particularity of electing words as labels, which may not appear in the cluster.

Finally, we propose a new evaluation methodology based on standard linguistic resources such as WordNet [8], Wikipedia (http://www.wikipedia.org) and the Britannica Encyclopedia (http://www.britannica.com). The idea is to map the different cluster labels and contents to existing well-known classifications of a query. As a consequence, the optimal ephemeral clustering methodology would be the one, which (1) assigns one and only one meaning/topic to a cluster and (2) a meaning/topic would only be embodied by a unique cluster. We will show that within this context, our methodology outperforms all state-of-the-art systems.

II. Related Work

Ephemeral clustering has been studied for more than a decade and many studies have been proposed as summarized in Table I. As stated in [9], there exist two main different approaches: monothetic clustering (also known as label-centered clustering) and polythetic clustering (also known as document-centered clustering). Monothetic algorithms are those in which a document is assigned to a cluster based on a single feature, whereas polythetic algorithms assign documents to the clusters based on multiple features.

Although the main difference between all approaches is the clustering strategy, many other characteristics can classify ephemeral clustering as shown in Table I. These are all exhaustively defined in [10] and [2], who clearly settle the foundations of post-retrieval document browsing. According to [2], *both cluster overlap and multi-word phrases are critical* to the success of their suffix tree

Table I
CLASSIFICATION OF EPHEMERAL CLUSTERING.

Work	Taxonomy	Algorithm	Overlap
[10]	Flat and Hierar.	Document-center.	No
[2]	Flat	Label-center.	Yes
[1]	Hierar.	Document-center.	No
[11]	Hierar.	Document-center.	No
[12]	Flat	Doc./Label-center.	Yes
[13]	Hierar.	Label-center.	No
[14]	Hierar.	Label-center.	No
[15]	Flat	Label-center.	Yes
[3]	Flat	Label-center.	Yes
[16]	Lattice	Label-center.	Yes
[9]	Hierar.	Label-center.	Yes
[17]	Hierar.	Document-center.	Yes
[4]	Hierar.	Label-center.	Yes
[18]	Hierar.	Document-center.	Yes
[5]	Flat	Label-center.	Yes

Work	Text	MWU	Labels
[10]	Document	No	No
[2]	Snippet	Yes	Yes
[1]	Document	No	Yes
[11]	Snippet	Yes	Yes
[12]	Snippet	No	No/Yes
[13]	Document	No	Yes
[14]	Document	Yes	Yes
[15]	Snippet	Yes	Yes
[3]	Snippet	Yes	Yes
[16]	Snippet	No	Yes
[9]	Snippet	Yes	Yes
[17]	Document	Yes	Yes
[4]	Snippet and KB	Yes	Yes
[18]	Snippet	Yes	Yes
[5]	Snippet	Yes	Yes

clustering algorithm (STC), which means that multiword units (MWU) best embody the message conveyed by texts as well as documents may belong to different clusters as they may focus on different topics. Moreover, they show that applying clustering algorithms based on the overall documents instead of their corresponding Web snippets leads to improved results both in the case of monothetic and polythetic strategies[1], except for the well-known K-means algorithm[2]. However, the decrease in quality of the clusters is apparent but relatively small. As a consequence, they argue that Web snippets are likely to provide the correct clustering of the documents as they embody the excerpts of the documents mostly related to the query terms.

Another important issue is stated in [10] and confirmed later in [1]. Indeed, [10] compare both flat and hierarchical clustering based on the classical vector space model with a partitioning algorithm called fractionation and show that *the best cluster is actually the result of two clustering steps (the best of the best)*. As a consequence, based on the statements of [10] and [2], ephemeral clustering should tackle normalized[3] Web snippet hierarchical overlapping clustering to produce relevant results. Within this scope, the best strategies have been proposed by [9] [4] and [18].

However, users still feel reluctant to use search engines implementing ephemeral clustering. We deeply believe that one of the main reasons is the fact that too many clusters are presented to the users who prefer to scan long lists of Web pages rather than going through long lists of (possibly misdescriptive) labels of clusters. For instance, to avoid this problem, [10] [2] and [9] respectively show the top 5, 10 and 5 clusters. Indeed, as stated in [9] and [15], since the main purpose of the taxonomy is to provide a better browsing experience, the taxonomy should be as compact as possible.

Based on this idea, we propose a document-centered solution, which (1) integrates the semantic dimension by using the InfoSimba measure [6] to evaluate the similarity between Web snippets, (2) proposes a new divisive hard clustering process based on a variant of the global K-means algorithm [7] to produce a hierarchy of compact concepts and (3) includes the selection of cluster labels within the clustering process. We call this algorithm the hierarchical InfoSimba-based global K-means ($HISGK$-means). Such, we aim at reaching query-based disambiguation by reducing the number of clusters and providing meaningful labels[4].

III. WEB SNIPPET REPRESENTATION

A. Web Snippet Segmentation

Web snippet segmentation must be done to be able to clearly understand the contents of Web snippets. In order to identify potential relevant text segments, most methodologies have been proposed in the context of label-centered algorithms. Most of them are based on the extraction of frequent sets of words that appear together in more than a minimum fraction of the whole document set. For that purpose, different approaches have been proposed. [2] implement a suffix tree structure, [11] and [3] propose a suffix-array methodology, [13] use association rules to extract itemsets, [15] learn a linear regression and [4] propose to extract common gapped sentences from linguistically enriched Web snippets. As one may want to search over the entire Web in any language, it is important that the clustering algorithm only depends on language-independent features. Within this scope, the identification of relevant text segments is mainly based on frequency of occurrence as the unique clue for extraction. However, this methodology suffers from the poor quality of Web snippets, which mainly contain ill-formed sentences with many repetitions. In order to avoid these problems, [5] proposed an interesting methodology based on three different metrics to analyze words distribution, which we will follow in our work.

B. Web Snippet Similarity

While existing methodologies, both polythetic or monothetic, evaluate the similarity between Web snippets based

[1]In this case, they use the classical vector space model.

[2]A result they cannot explain.

[3]With the identification of MWU.

[4]The overlapping version of the $HISGK$-means and the identification of MWU are out of the scope of this paper.

on the exact match of constituents, we propose that two Web snippets are highly related if both share highly related (eventually different) constituents. So, similarity is not any more based on the exact match of constituents but on related words. Indeed, it is clear that both sentences (1) and (2) are similar although they do not share any word.

(1) *Ronaldo defeated the goalkeeper once more.*
(2) *Real Madrid striker scored again.*

This situation can easily be understood as *Ronaldo* from sentence (1) is highly correlated to *Real Madrid*, *striker* etc. from sentence (2). The InfoSimba similarity measure proposed in [6] models this phenomenon in an elegant way. Within the polythetic strategy, each Web snippet is represented by a vector of its most relevant words i.e. the set of the best p words selected based on the lowest $K(.)$ scores from [5]. So, given two Web snippets X_i and X_j, their similarity is evaluated by the simplified InfoSimba measure defined in Equation 1 where $S(.,.)$ is any symmetric similarity measure and each W_{ij} corresponds to the word at the j^{th} position in the vector X_i and X_{ij} corresponds to the weight of the word W_{ij}.

$$ISs(X_i, X_j) = \frac{1}{p^2} \sum_{k=1}^{p} \sum_{l=1}^{p} X_{ik}.X_{jl}.S(W_{ik}, W_{jl}). \quad (1)$$

In particular, we will use the Symmetric Conditional Probability association measure $SCP(.,.)$ and defined in Equation 2 to evaluate the correlation between two word vector constituents i.e. $S(.,.)$, where $P(.,.)$ is the joint probability of two words appearing in the same Web snippet and $P(.)$ is the marginal probability of any word appearing in a Web snippet.

$$S(.,.) = SCP(x,y) = \frac{P(x,y)^2}{P(x) \times P(y)}. \quad (2)$$

IV. The $HISGK$-means Algorithm

The main goal of our approach is to hierarchically organize Web snippets into a compact taxonomy, guaranteeing that at each level of the hierarchy, we automatically find the most suitable number of clusters and extract for each cluster a small set of representative words (i.e. the cluster labels). For that purpose, we propose a new hierarchical divisive hard clustering algorithm called the hierarchical InfoSimba-based global K-means ($HISGK$-means) algorithm for which we provide a well-founded mathematical background, which guarantees optimal clustering. In particular, the hierarchical process is a top-down approach, which recursively splits a set of Web snippets based on a variant of the global K-means algorithm (GK-means) [7] combined with the simplified InfoSimba informative similarity measure, which we call the InfoSimba-based global K-means ($ISGK$-means). The procedure is defined in algorithm 1.

In order to understand all the procedure, we first need to describe the well-known K-means algorithm and its adaptation to the InfoSimba similarity measure. The K-means method is a well known geometric clustering algorithm based on work by [19]. Given a set of n data points, the algorithm uses a local search approach to partition the points into K clusters. A set of K initial cluster centers is chosen. Each point is then assigned to its closest center and the centers are recomputed as centers of mass of their assigned points. This is repeated until the process stabilizes. It can be shown that no partition occurs twice during the course of the algorithm, and so the algorithm is guaranteed to terminate. In order to assure convergence, an objective function Q must be defined, which decreases at each step of the algorithm.

Algorithm 1 The $HISGK$-means algorithm

Input: A set of Web snippets S and a stop criterion C
Output: A hierarchy
Initialize the root h_0 of the hierarchy to S
Initialize the level of the hierarchy to 1 i.e. $l = 1$
Initialize the number of representative words for the centroid to 2 i.e. $p = 2$
Apply $ISGK$-means at level h_0
Retrieve K_0 clusters $h_{1,1}, \ldots, h_{1,K_0}$
Link all clusters $h_{1,k}$ to their parent h_0
Label all clusters $h_{1,k}$ and h_0 based on their p-sized centroids
$l = l + 1$
$p = p + 1$
for Each cluster $h_{l-1,k}$ and C is true **do**
  Apply $ISGK$-means at level h_{l-1}
  Retrieve K_l clusters $h_{l,1}, \ldots, h_{l,K_l}$
  Link all clusters $h_{l,k}$ to their parent h_{l-1}
  Label all clusters $h_{l,k}$ and h_{l-1} based on their p-sized centroids
  $l = l + 1$
  $p = p + 1$
end for

In the particular context of Web snippets clustering, the K-means algorithm needs to be adapted in order to use the InfoSimba similarity measure. Indeed, a Web snippet is not defined by a numerical vector but by a set of p words (i.e. a word context vector of size p) over which a proximity coefficient is defined, in this case, the simplified InfoSimba $ISs(.,.)$ defined in Equation 1. In particular, all words contained in the word context vector are given a score of 1. As a consequence, we define the objective function Q_{IS} to maximize during the clustering process in Equation 3.

$$Q_{IS} = \sum_{k=1}^{K} \sum_{x_i \in \pi_k} ISs(x_i, m_{\pi_k}). \quad (3)$$

Notice that a cluster centroid m_{π_k} is now defined by a p-

context vector of words $(w_1^{\pi_k}, \ldots, w_p^{\pi_k})$. As a consequence, we must define a way to update cluster centroids in such a way that Q_{IS} increases at each step of the clustering process. The choice of the best p words representing each cluster is a way of assuring convergence. For that purpose, we define the procedure UPDATE(π_k), which consists in selecting p words from the global vocabulary V in such a way that Q_{IS} is improved. The global vocabulary is the set of all words, which appear in any context vector[5]. So, for each word $w \in V$ and any proximity coefficient PC (in this case, the SCP association measure), we compute its interestingness $\lambda^k(w)$ as regards to cluster π_k as defined in Equation 4 where $s_i \in \pi_k$ is any Web snippet from cluster π_k and only select the p words with higher interestingness value to construct the cluster centroid. We can easily show that Q_{IS} is maximized in such a way and assures convergence.

$$\lambda^k(w) = \frac{1}{p} \sum_{s_i \in \pi_k} \sum_{w_q^i \in s_i} PC(w_q^i, w). \quad (4)$$

So, the adaptation of the K-means is straightforwardly defined in algorithm 2 and called the InfoSimba-based K-means (ISK-means).

Algorithm 2 The ISK-means algorithm

Input: Number of K, a set of Web snippet X, List of Centroids L_{in}
Output: K partitions, List of Centroids L_{out}
Initialize K cluster centers in X, randomly and/or using L_{in}
while convergence is not obtained **do**
  Assign each Web snippet $s_i \in X$ to its nearest cluster using $ISs(.,.)$
  Update each cluster center by computing its centroid using UPDATE(π_k)
end while

Now that the well-known K-means has been adapted to the case of Web snippet clustering, we introduce the GK-means clustering algorithm [7], which is at the basis of our overall $HISGK$-means algorithm. The GK-means constitutes a deterministic effective global clustering algorithm for the minimization of the clustering error that employs the K-means algorithm as a local search procedure. The algorithm proceeds in an incremental way. As such, to solve a clustering problem with M clusters, all intermediate problems with $1, 2, ..., M-1$ clusters are sequentially solved. The basic idea underlying the proposed method is that an optimal solution for a clustering problem with M clusters can be obtained using a series of local searches using the classical K-means algorithm. At each local search, the $M-1$ cluster centers are always initially placed at their optimal positions corresponding to the clustering problem with $M-1$ clusters. The remaining M^{th} cluster center is initially placed at several positions within the data space. Since for $M = 1$ the optimal solution is known, it is possible to iteratively apply the above procedure to 2^{nd} optimal solutions for all K-clustering problems $K = 1, ..., M$. In addition to effectiveness, the method is deterministic and does not depend on any initial conditions or empirically adjustable parameters. Moreover, its adaptation to the specific case of Web snippet clustering is direct as shown in algorithm 3. We call this algorithm the InfoSimba-based global K-means ($ISGK$-means).

Algorithm 3 The $ISGK$-means algorithm

Input: Number of K, a set of Web snippets X
Output: K partitions, List of Centroids L_{out}
Run ISK-means(1, X, [])
$Lcentroids_1 \leftarrow$ centroid of ISK-means(1, X, [])
for Each $k = 2$ to $k = K$ **do**
  Run ISK-means(k, X, $Lcentroids_{k-1}$)
  $Lcentroids_k \leftarrow$ centroids of ISK-means(k, X, $Lcentroids_{k-1}$)
end for

Once the clustering process has been handled, selecting the best number of clusters still remains to be decided. In most real life clustering situations, selecting the number of clusters in the final solution is a hard and still opened problem. Usually the user requires to define *a priori* the desired number of clusters. As a consequence, numerous procedures to determine the "best" number of clusters dividing a data set have been proposed [20]. However, none of the listed procedures were effective or adaptable to our specific problem. As a consequence, we proposed a new methodology based on the definition of a rational function, which models the quality criterion Q_{IS} in the context of the $ISGK$-means algorithm. Although this issue is out of the scope of this paper, the basic idea is to model Q_{IS} with a rational function, which converges.

The $HISGK$-means shows interesting properties. First, it is mathematically well-founded so that optimum clustering is guaranteed. Second, the labeling step is embodied in the clustering process, thus avoiding an extra step to label clusters, unlike all polythetic strategies proposed so far. Indeed, the cluster centroid is chosen as the label of the cluster. In particular, the best label terms for a given cluster can be words outside the cluster itself. Third, it is applied on a language-independent architecture as stop-words removal, stemming, lemmatization or linguistic resources are not used or applied contrarily to most of the existing systems. And fourth, it retrieves compact taxonomies as opposed to start-of-the-art methodologies.

[5]Notice that V can be high.

V. EVALUATION

Evaluating ephemeral clustering is not an easy task. In particular, two different approaches have been followed. On the one hand, some works propose automatic evaluations such as in [10] [2] [1] [13] and [14]. In particular, [10] [2] and [13] propose a classical information retrieval evaluation where they compare manually annotated sets of documents or Web snippets with ephemeral clusters obtained for a given query. Unfortunately, they use different data sets thus preventing possible direct comparisons. Moreover, only a small number of queries are used for evaluation e.g. 49 for [10] and 10 for [2]. In [13], the evaluation process is even limited to the clustering of documents without a given query. Instead, [1] and [14] respectively propose to evaluate the quality of the generated taxonomy based on information theoretic measures and statistical tests. One major drawback of this methodology is that they depend on the length of the hierarchy. For that purpose, they only use a small number of documents e.g. 1700 documents and Web snippets for [14] and 430 documents for [1].

On the other hand, some works propose to perform user studies such as in [12] [15] [3] [9] and [4]. Their objective is to prove that presenting interface facilities can improve retrieval speed or user satisfaction. Although these studies are important, they are difficult to reproduce and their results may be subjective mainly depending on the audience (e.g. young vs. old users, advanced vs. not advanced users etc.), the clarity of the interface (e.g. aesthetics, user experience), the methodology (e.g. effectiveness, efficiency and satisfaction defined in ISO 9241-11) or the complexity of queries (e.g. general language vs. specialized language) to name but a few. Moreover, due to the tedious task of manual evaluation, only very few queries can be tackled.

Only very few works proposed comparative evaluations between different algorithms. [2] [12] and [9] are the few exceptions. [2] compared their STC algorithm to the one proposed in [10] and four other well-known clustering algorithms (single-pass, K-means, buckshot, GAHC). [12] proposed a visual comparison with the STC algorithm and [9] compared their Discover algorithm with their previous CAARD [21] and the DSP algorithm proposed in [14].

It is clear that great efforts must be carried out to assess the quality of existing algorithms by proposing automatic evaluations on golden standards. Indeed, although some comparative studies have been proposed in terms of evaluation, they are usually based on a small set manually labeled documents as well as confronted to a small number of queries. Our objective is clearly different from all evaluation schemes proposed so far. Indeed, in the optimal case, ephemeral clustering should be able to automatically discover the different meanings of the query terms. Even more, the optimal ephemeral clustering methodology would be the one, which (1) assigns one and only one meaning/topic to a cluster and (2) a meaning/topic would only be embodied by a unique cluster. Moreover, to assess the reliability of such systems, evaluation should be performed on a large number of queries in a real-world environment i.e. the Web.

For that purpose, we automatically extracted from the AOL query log data set (http://www.gregsadetsky.com/aol-data), which consists of 10,154,742 unique Web queries collected from 657,426 users between 1st March 2006 and 31st May 2006, all the queries, which are present at the same time in WordNet [8], Wikipedia and the Britannica Encyclopedia. This extraction resulted in 1,419 queries. For each query, we then automatically extracted all their different topics or meanings i.e. (1) the different synsets and respective glosses within WordNet, (2) the different categories and sub-categories from the disambiguation facility of Wikipedia and (3) the results from the Britannica Encyclopedia. As a consequence, our objective is to study on a large scale how well existing methodologies can retrieve the maximum number of meanings or topics reported in well-known databases, but at the same time do not over-generate clusters. As a consequence, we propose four different metrics Δ_1, Δ_2, Δ_3 and Δ_4 respectively defined in Equations 5, 6 and 7, 8 to evaluate the taxonomy compactness.

$$\Delta_1 = \frac{\sum_{\text{all meanings}} \text{\# of associated clusters}}{card(\text{all meanings})}. \quad (5)$$

$$\Delta_2 = \frac{\sum_{\text{all clusters}} \text{\# of associated meanings}}{card(\text{all clusters})}. \quad (6)$$

$$\Delta_3 = \frac{\sum_{\text{all queries}} \frac{\text{\# of found meanings}}{\text{\# of generated clusters}}}{card(\text{all queries})}. \quad (7)$$

$$\Delta_4 = \frac{\sum_{\text{all queries}} \frac{\text{\# of found meanings}}{\text{\# of existing meanings}}}{card(\text{all queries})}. \quad (8)$$

The objective of Δ_1 is to identify how many clusters are linked to a given meaning or topic of the query on average. Oppositely, Δ_2 evaluates how many meanings or topics are linked to a given generated cluster on average. As such, if Δ_1 is high, it means that a given meaning/topic is associated to many clusters, which jeopardizes the compactness of the taxonomy. Oppositely, if Δ_2 is high, it means that the generated clusters embody many different meanings and as such prejudice the quality of the clustering. As a summary, the optimal system would be the one that produces the smallest average of Δ_1 and Δ_2 i.e. $\frac{\Delta_1+\Delta_2}{2}$. Finally, Δ_3 evidences the compactness of the generated taxonomy i.e. the higher Δ_3, the more compact the taxonomy will be, and

Δ_4 evidences the coverage of the taxonomy i.e. the higher Δ_4, the more complete the taxonomy will be.

In order to compute both Δ_1, Δ_2, Δ_3 and Δ_4 it is necessary to automatically associate meanings/topics to clusters. For that purpose, we propose to use the classical cosine similarity measure defined in Equation 9 between the word vector associated to each cluster and the word vector associated to each definition from the knowledge base i.e. WordNet, Wikipedia and the Britannica Encyclopedia.

$$cos(X_i, X_j) = \frac{\sum_{k=1}^{p} X_{ik} \times X_{jk}}{\sqrt{\sum_{k=1}^{p} X_{ik}^2} \times \sqrt{\sum_{k=1}^{p} X_{jk}^2}}. \quad (9)$$

It is important to notice that the word vectors depend on (1) the systems being studied and (2) the knowledge base. In this first experiment, we will compare the CBL algorithm [5], the Yippy search engine and the $HISGK$-means algorithm. While the Web services of the CBL and the $HISGK$-means retrieve the labels plus a number of related words, the Yippy Web service only allows the access to the cluster labels. As such, the clusters generated by the CBL and the $HISGK$-means algorithms will be represented by longer representative word vectors compared to Yippy. As a consequence, to propose a fair evaluation, we will only compare Yippy to the labels of the CBL and the $HISGK$-means algorithms[6]. It is interesting to notice that such a way we are able to compare both the quality of the clustering as well as the quality of the labeling.

Similarly, each meaning of a query can be represented in WordNet by its synonyms and its gloss. As a consequence, we will evaluate the similarity between the clusters and the meanings in WordNet based on their full representation (i.e. synonyms plus gloss) and on their short representation (i.e. synonyms), which can be assimilated to the capacity of the ephemeral clustering algorithm to find correct labels. The same will be true for Wikipedia and its word sense disambiguation facility. However, the Britannica Encyclopedia only proposes a single word to express the different meanings. As such, smaller representative word vectors will be available.

In Tables II III IV, we illustrate the results of our evaluation. In particular, we show the results for the best combination of p and K for the $HISGK$-means algorithm.

The results of our automatic evaluation are clear. For the full representation, the $HISGK$-means algorithm systematically proposes a smaller Δ_1 compared to CBL. In particular, it is illustrated by the fact that each meaning/topic tends to be covered by 1.5 clusters on average by the $HISGK$-means while the CBL algorithm evidences that on average 5.76 clusters cover the same meaning. Oppositely, it is obvious that each one of the clusters generated by CBL tends to focus on less meanings/topics. However, the distance in terms of Δ_2 between the CBL and the $HISGK$-means is low, which makes us conclude that the $HISGK$-means is able to generate much less ambiguous clusters and at the same time manages to clearly separate between different meanings. This statement is supported by a systematic low quotient $\frac{\Delta_1+\Delta_2}{2}$. Moreover, by analyzing Δ_3 and Δ_4, it is clear that the $HISGK$-means is capable of covering most of the meanings encountered by CBL but generates much less clusters as illustrated by an average of the quotient $\frac{\Delta_3+\Delta_4}{2}$ of 0.9 for the $HISGK$-means and 0.48 for the CBL.

Exactly the same situation occurs for the experiments based on the label description for WordNet and Wikipedia, thus validating the quality of the labeling of the $HISGK$-means algorithm compared to CBL and Yippy. In particular, the CBL algorithm outperforms the Yippy search engine

[6]At this stage, the algorithms of [2], [3] and [4] have not been included yet in the evaluation as they do not offer Web services.

Table II
WORDNET RESULTS USING FULL AND LABEL DESCRIPTION.

WORDNET					
Algorithm	Δ_1	Δ_2	$\frac{\Delta_1+\Delta_2}{2}$	Δ_3	Δ_4
Full Description					
CBL	6.38	1.44	3.91	0.12	0.68
$HISGK$-means (p=8, K=8)	1.50	1.60	1.55	0.49	0.61
Label Description					
Yippy	1.60	1.17	1.39	0.03	0.47
CBL	1.49	1.22	1.35	0.06	0.44
$HISGK$-means (p=8, K=8)	1.17	1.36	1.27	0.27	0.39

Table III
WIKIPEDIA RESULTS USING FULL AND LABEL DESCRIPTION.

WIKIPEDIA					
Algorithm	Δ_1	Δ_2	$\frac{\Delta_1+\Delta_2}{2}$	Δ_3	Δ_4
Full Description					
CBL	6.09	1.35	3.72	0.14	0.58
$HISGK$-means (p=6, K=8)	1.58	1.43	1.51	0.59	0.53
Label Description					
Yippy	1.61	1.07	1.34	0.06	0.48
CBL	1.50	1.10	1.30	0.08	0.37
$HISGK$-means (p=6, K=8)	1.23	1.19	1.21	0.33	0.30

Table IV
BRITANNICA RESULTS USING FULL AND LABEL DESCRIPTION.

BRITANNICA					
Algorithm	Δ_1	Δ_2	$\frac{\Delta_1+\Delta_2}{2}$	Δ_3	Δ_4
Full Description					
CBL	4.82	5.48	5.15	0.78	0.31
$HISGK$-means (p=4, K=14)	1.45	5.53	3.50	2.91	0.28
Label Description					
Yippy	1.22	2.60	1.91	0.28	0.22
CBL	1.13	3.09	2.11	0.44	0.18
$HISGK$-means (p=8, K=14)	1.11	3.89	2.50	1.41	0.14

within this context. However, different results are obtained for the Britannica Encyclopedia. This is mainly due to its structure as instead of describing different query meanings, the Britannica Encyclopedia outputs related words, which most of the time intersect in meaning. As a consequence, although Δ_1 shows its lowest value for the $HISGK$-means algorithm, the corresponding high Δ_2 value evidences that each cluster embodies many "different" meanings. In fact, these meanings are not so different. For example, for the query *jaguar*, the Britannica Encyclopedia would differentiate between the meanings of *plant and animal life from the articlenicaragua* and *plant and animal life from the articleguatemala*. As a consequence, it is not strange to find many overlapping meanings by cluster as well as Δ_3 values superior to 1. In fact, the intrinsic structure of the Britannica tends to benefit the over-generation of clusters.

VI. Conclusions

In this paper, we proposed a new algorithm for ephemeral clustering, the $HISGK$-means. In particular, it builds a compact hierarchical taxonomy in a language-independent framework as no linguistic information is introduced, thus keeping to the *corpus integrity principle*. The automatic evaluation framework showed that the $HISGK$-means outperforms the Yippy and the CBL algorithm in terms of compactness and ambiguity. To some extent, we proposed a methodology, which approximates the optimal ephemeral clustering i.e. the one, which (1) assigns one and only one meaning/topic to a cluster and (2) a meaning/topic would only be embodied by a unique cluster. However, although stimulating results are obtained, there is still much to improve about the overall methodology. In particular, to answer the statements of [10] and [2], we will tackle the identification of MWU and propose an overlapping version of the $HISGK$-means, in the near future.

References

[1] Y. Maarek, R. Fagin, I. Ben-Shaul, and D. Pelleg, "Ephemeral document clustering for web applications," IBM, Tech. Rep., 2000.

[2] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration," in *the 21st ACM SIGIR Conference*, 1998, pp. 46–54.

[3] S. Osinski, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *the IIPWM Conference*, 2004, pp. 369–378.

[4] P. Ferragina and A. Gulli, "A personalized search engine based on web-snippet hierarchical clustering," *Software: Practice and Experience*, vol. 38, no. 2, pp. 189–225, 2008.

[5] D. Machado, T. Barbosa, S. Pais, B. Martins, and G. Dias, "Universal mobile information retrieval," in *Proceedings of the 13th International Conference on Human Computer Interaction (HCII 2009)*, 2009.

[6] G. Dias, "Information digestion," 2010, university of Orléans.

[7] A. Likasa, V. N., and J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, pp. 451–461, 2003.

[8] G. A. Miller, "Wordnet: an on-line lexical database," *International Journal of Lexicography*, vol. 3, no. 4, 1990.

[9] R. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram, "A hierarchical monothetic document clustering algorithm for summarization and browsing search results," in *the 13th WWW Conference*, 2004, pp. 658–665.

[10] M. Hearst and J. Pedersen, "Re-examining the cluster hypothesis: Scatter/gather on retrieval results," in *the 19th ACM SIGIR Conference*, 1996, pp. 76–84.

[11] D. Zhang and Y. Dong, "Semantic, hierarchical, online clustering of web search results," in *the 6th APWEB Conference*, 2001, pp. 69–78.

[12] Z. Jiang and A. Joshi, "Retriever improving web search engine results using clustering," *Managing Business with Electronic Commerce: Issues and Trends*, pp. 59–81, 2002.

[13] B. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets," in *the 3rd SDM Conference*, 2003, pp. 59–70.

[14] D. Lawrie and B. Croft, "Generating hierarchical summaries for web searches," in *the 26th ACM SIGIR Conference*, 2003, pp. 457–458.

[15] Q. Zeng, Q. He, C. Zheng, and J. Ma, "Learning to cluster web search results," in *the 27th ACM SIGIR Conference*, 2004, pp. 210–217.

[16] C. Carpineto and G. Romano, "Exploiting the potential of concept lattices for information retrieval with credo," *Journal of the Universal Computer Science*, vol. 10, no. 8, pp. 985–1013, 2004.

[17] R. Campos and G. Dias, "Automatic hierarchical clustering of web pages," in *Proceedings of the ELECTRA Workshop of the 28th Annual International ACM SIGIR Conference (SIGIR 2005)*, 2005, pp. 83–85.

[18] G. Dias, S. Pais, F. Cunha, H. Costa, D. Machado, T. Barbosa, and B. Martins, "Hierarchical soft clustering and automatic text summarization for accessing the web on mobile devices for visually impaired people," in *Proceedings of the 22nd Florida Artificial Intelligence Research Society Conference (FLAIRS 2009)*, 2009.

[19] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[20] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[21] R. Kummamuru and R. Krishnapuram, "A clustering algorithm for asymmetrically related data with its applications to text mining," in *the 10th CIKM Conference*, 2001, pp. 571–573.