Exploiting Web Sites Structural and Content Features for Web Pages Clustering

***Abstract*—Web page clustering is a focal task in Web Mining to evaluate interactions among web pages, organize the content of websites and understand their structure. It can be applied into several research areas (e.g. entity discovery, entity linking, information retrieval, etc.). However, it is a tricky task since web pages are characterized by several representations based on textual, hyperlink and HTML formatting (i.e. HTML tags and visual) properties. Existing clustering algorithms use these information almost independently, mainly because it is difficult to combine them. This paper is intended to be a contribution on clustering of web pages in a website by combining all this features into a single vector space representation. In particular, the proposed approach first crawls the website by using web pages' HTML formatting and exploiting *web list* in order to identify and represent the hyperlink structure by means of an adapted skip-gram model. Then, this hyperlink structure is enriched with the content information into a single vector space representation which can be used by any traditional and best-performing clustering algorithms. An experimental evaluation on several websites shows that combining these heterogeneous information in a single representation can improve the effectiveness of the clustering task.**

1. Introduction

The process of automatically organizing web pages and websites has always attracted extensive attention in several research areas because of its significant theoretical challenges and because of its great application and commercial potentials. Clustering represents one of the most investigated techniques used to evaluate the interaction among web pages and organize them into groups such that web pages within the same group are more similar to each other than those belonging to different ones. As a consequence, clustering algorithms can be profitably used to organize the content of websites and to understand their hidden structure or, in Information Retrieval, to provide an effective representation of search engine results [1].

Since a web page is characterized by several representations (i.e. textual representation, structural representation based on HTML tags and visual features and structural representation based on hyperlinks) the existing clustering algorithms differ in their ability of using these representations.

In particular, clustering based on textual representation typically group web pages using words distribution [1], [2], [3]. Solutions based on this approach manage web pages as plain text corpora ignoring all the other information of which is enriched. These algorithms turn to be ineffective in at least two cases: *i)* when there is not enough information in the text of a page; *ii)* when websites have different content, but actually refer to the same semantic class. The former case refers to web pages of poor of textual information, such as pages rich of structural data (e.g. pages from Deep Web Databases) or multimedia data, or when web pages have several script terms, which can be easily found also in other pages (e.g. pages from a CMS website). The latter case refers to web pages having the same semantic type (e.g. web pages related to professors, courses, books, lists of publications of a single researchers) but characterized by different distribution of terms.

On the other side, clustering based on structure typically considers the HTML formatting (i.e. HTML tags and visual information rendered by a web browser) [4], [5], [6]. Algorithms, which use these information to organize web pages, are based on idea that web pages are automatically generated by programs that extract data from a back-end database and embed them into an HTML template. Web pages generated with this approach, show a common structure and layout, while differing in content. However, because tags are more often used for styling purposes than for content structuring, it can happen that most web pages in a website have the same structure, even if they refer to distinct semantic types. The effect is that the clustering algorithm will generate very few low-quality clusters.

These two solutions, however, only exploit within-page information. On the contrary, other solutions also exploit the graph defined by the hyperlink structure of a set of web pages [7], [8]. Clustering on hyperlink structure is based on idea that web pages, differently from traditional textual documents, are enriched by hyperlinks that enable information to be split in multiple and interdependent web pages. These hyperlinks can be used to identify collections of web pages semantically related and relationships among these collections. For example, the website of a computer science department may organize its web pages in collections of courses, research areas, and people; people may be further split into faculty, staff and students. Analyzing the link structure allow us to discover the hidden structure of website, which can be used to aid the navigation and the understanding of a website. However the effectiveness of clustering algorithms strongly depends by links density and presence of noisy links.

Although there are several works that focus on web pages clustering, most of them analyze contents, web page structure (i.e. HTML formatting) and hyperlink structure of a website almost independently, mainly because different sources of information use different data representations. Over the last decade, some researchers tried to combine different sources of information. For example, [7], [9], [10], [11], [12], [13] combine content and hyperlink structure for web page clustering, [8], [14], [15], [16], [17] combine content and hyperlink structure for classification and [4] combines web page and hyperlink structure for clustering purposes.

This paper is intended to be a contribution in the direction of performing clustering of web pages in a website by combining information about content, web page structure and hyperlink structure of web pages. This goal is achieved analyzing web pages' HTML formatting to extract from each page collections of links, called *web lists*, which can be used generate a compact and noise-free representation of the website's graph. Then, the extracted hyperlink structure and content information of web pages are mapped in a single vector space representation which can be used by any traditional and best-performing clustering algorithms.

Specifically, our approach is based on the idea that two web pages are similar if they have common terms (i.e. *Bag of words hypothesis* [18]) and they share the same reachability properties in the website's graph. In order to consider reachability, the solution we propose is inspired by the concept of Distributional hypothesis, initially defined for words in natural language processing (i.e. "*You shall know a word by the company it keeps*") [19]and recently extended to generic objects [20]. In the context of the Web we can translate that citation in "*You shall know a web page by the paths it keeps*". According to this hypothesis two web pages are similar if they are involved in the same paths in the website's graph.

This paper is organized as follows: In the next section we describe some related work on clustering of web pages. In Section 3 we present our solution which is empirically evaluated in Section 4. We conclude with main findings of the paper and future work in Section 5.

2. Related Work and Motivations

Many techniques have been used for clustering large corpora of web pages. In particular, existing algorithms are based on two main approaches: *content analysis* and *structure analysis*.

As clarified before, the most important clustering algorithms based on **web pages contents** treat web pages as independent textual documents. This is the case of [1], [2], [3], [21], where the words distribution is used to discover appropriate groups of interrelated web pages. The advantage of this approach is that many off-the-shelf clustering tools based on vector space model can be directly applied. However, these approaches fail to learn accurate models due the uncontrolled and heterogeneous nature of web page contents. In fact, traditional clustering algorithms are based on the assumption that textual documents share consistent writing styles, provide enough contextual information, are plain and completely unstructured, and are independent and identically distributed. These limitations are more obvious for clustering of web pages belonging to different websites or whose content is created in a collaborative manner. In this case, in fact, web pages related to the same topic could be contextually different, contain similar information into web elements with different semantic rules (e.g., navigational menu, main content, calendar, tables, and logotypes) and different functionalities (e.g., links, buttons, images, anchor-texts, etc.) [22].

On the contrary of plain text documents, web pages are characterized by structural properties such as HTML tags, visual features and hyperlinks which define the structural representations of web pages. It as been proved (see [4], [6], [7], [17]) that such structural information provide different and complementary information respect to a textual representation.

A different perspective is represented by clustering algorithms that exploit structural properties. They can be organized in two main categories: *i)* clustering based on HTML formatting and *ii)* clustering based on hyperlink structure. Clustering based on **HTML formatting** takes advantage of the structural and visual information embedded in HTML tags, which are usually ignored by plain text approaches. In the literature several works (see, for instance [5], [23]) propose to organize web pages according with their structural information. They are mainly based on the assumption that HTML documents can be treated as XML documents. However, differently from XML tags, HTML tags are oriented towards data visualization rather than data representation. The consequence is that it can appear that web pages of the same semantic type (e.g., web pages of professors) are codified and rendered using different tags structures or, viceversa, different types of web pages are codified with same tags structure. For example, structured data can be coded with a HTML table (<table>) tag or HTML list (<ul>) and have similar appearance. This defaces the quality of generated clusters.

To overcome this limit, in [4], [6], the authors proposed to use visual information associated with HTML tags. Specifically, in [6] clustering is based only on visual properties of web pages. Goal of this approach is to group web pages that could not have similar content, could not share the same HTML structure, but just look visually similar. In [4] layout and visual properties associated with HTML tags are used to characterize the structure of the whole web page, and collections of hyperlinks in a web page are used to find pages with similar structural representation.

Although visual information and HTML tags can improve the quality of web page clustering, algorithms based on HTML formatting work well for well-structured and homogeneous web pages. This limits their usage to web pages belonging to the same website and automatically generated from a back-end database (e.g., Deep Web Databases) or generated from dedicated content management systems (CMSs).

Clustering algorithms based on the **hyperlink structure** work on a interrelated collection of web pages. The basic idea is that when two web pages are connected via a link, there is some semantic relationship between them, which can be the basis for the partitioning of the collection into clusters. In general, such methods (see [24]) only use direct links among web pages and, on the base of them, they use/define some similarity measures (e.g., bibliographic coupling [25], co-citation [26], etc.). However, as claimed in [27], algorithms based on the hyperlink structure work well when the hyperlink structure is dense and noise-free. In fact, clustering based on the hyperlink structure returns low quality results for web pages without sufficient amount of in-links or out-links. Moreover, not all the links are equally important for the clustering process: web pages are often enriched with noisy hyperlinks such as hyperlinks used to enforce the web page authority in a link-based ranking scenario or with short-cut hyperlinks. To overcome these issues several algorithms combine content information with link information (see [7], [9], [10], [11], [12], [13]).

In this context, the earliest methods (for web page clustering) that try to combine textual information with co-citation and bibliographic coupling measures are [10], [11], [12]. Additionally, [9] faced the problem of noisy links, considering only hyperlinks among topically similar web pages and co-cited web pages. In particular, it associates a weight which combines the content similarity and co-citation to each edge (i, j) connecting the web pages i and j in the website graph. Afterwards, a traditional clustering algorithm based on graph partitioning is adopted. The method has two main limitations: $i)$ textual information are used only to weight links, then two web pages sharing same distributional properties but having low textual similarity cannot be clustered together; $ii)$ the graph clustering algorithm is NP-hard. In [13] a relaxation-labeling-based algorithm which first clusters documents based on their contents and then re-assigns the computed labels using the hyperlink structure among immediate neighbors is developed. To avoid the potential increase in the level of noise, only a subset of good neighbors are considered.

An alternative approach to better organize web pages belonging to the same website and remove noisy links is to filter link collections having similar visual and/or structural properties [4], [7], [8], [28], [29]. However, in our knowledge only [4] and [7] use information in link collections to improve clustering results. In [7] the authors propose a similarity measure obtained combining textual similarity, co-citation and bibliography-coupling similarity and *in-page link-structure* similarity. In this way, two web pages have a similar in-page link-structure if they appear more time together in a link collection. However, each measure needs a different representation space and their combinations is an open issue.

Previous solutions consider only direct relationships among neighbors, without analyzing the global structure of the website graph. [20], [30], [31], [32] claim that the similarity between two graph's nodes can be expressed in terms of similarity of their respective contexts, that is, how they share surrounding nodes (which could not necessary immediately neighbors). In [30], the authors propose a graph clustering algorithm that focuses on topological structure of a network and node properties, which can be textual or relational. A set of attribute nodes and attribute edges are added to the original graph. With such graph augmentation, the attribute similarity is transformed to vertex vicinity in the graph: two vertices which share an attribute value are connected by a common attribute vertex. Then, a neighborhood random walk model, which measures the vertex closeness on the augmented graph through both structure edges and attribute edges, unifies the two similarities. Although the algorithm is able to combine structural and content information using a common representation, it cannot be applied to data having numerical values (e.g. tf-idf) or categorical attributes with a huge amount of distinct values.

In [31], [32] two embedding methods are proposed (DeepWalk and Line, respectively). They exploit neural networks to generate a low-dimensional and dense vector representation of graph's nodes. DeepWalk [32] applies the skip-gram method on truncated random walks to encode long-range influences among graph's nodes. However, this approach is not able to capture the local graph structure (i.e. nodes which can be considered similar because are strongly connected). Line [31] optimizes an objective function that incorporates both the local (i.e. direct neighbors) and global (i.e. neighbors of neighbors) network structures. However, while DeepWalk is able to consider influences of arbitrary length, Line is able only to capture influence of length two. Moreover, both methods ignore node attributes (e.g. textual content). Consequently, clustering based on generated embedding might be difficult in graphs without sufficient in-links or out-links, but characterized by rich textual contents.

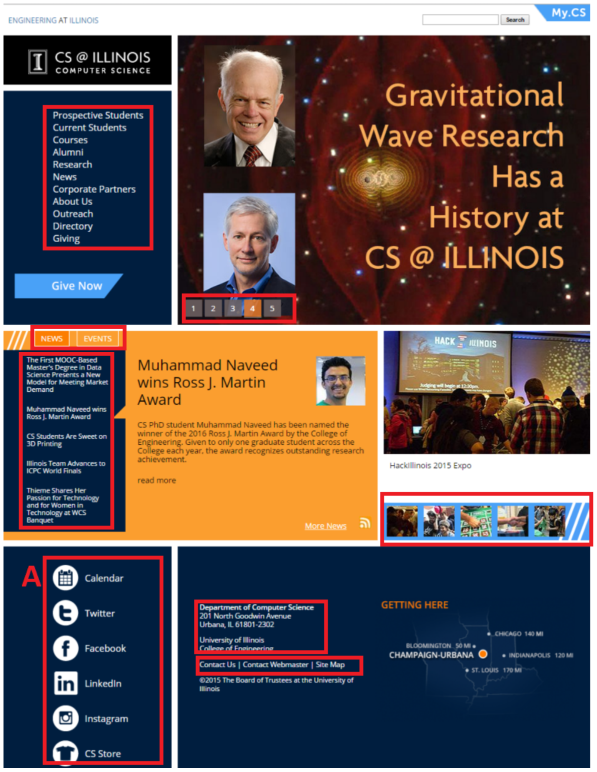
3. Methodology

As anticipated, the problem we consider is that of clustering web pages belonging to the same website by combining content, web page structure and hyperlink structure in a vector space representation. To achieve this goal, the proposed solution implements a four steps strategy: in the first step website crawling is performed. Crawling uses web pages' structure information and exploits web lists in order to mitigate problems coming from noisy links. The output of this phase is the website graph, where each node represents a single page and edges represent hyperlinks. In the second step, we generate a link vector by exploiting Random Walks extracted from the website's graph. In the third phase content vectors are generated. In the last phase, a unified representation of pages is generated and clustering is performed on such representation.

3.1. Website crawling

A Website can be formally described as a direct graph $G = (V, E)$, where V is the set of web pages and E is the set of hyperlinks. In most cases, the homepage h of a website representing the website's entry page allows the website to be viewed as a rooted directed graph.

As claimed in [4] not all links are equally important to describe the website structure. In fact, a website is rich of noisy links, which may not be relevant to clustering process,


ENGINEERING AT ILLINOIS
Search
My.CS
CS @ ILLINOIS
COMPUTER SCIENCE
Prospective Students
Current Students
Courses
Alumni
Research
News
Corporate Partners
About Us
Outreach
Directory
Giving
Give Now
Gravitational Wave Research Has a History at CS @ ILLINOIS
1
2
3
4
5
NEWS
EVENTS
The First MOOC-Based Master's Degree in Data Science Presents a New Model for Meeting Market Demand
Muhammad Naveed wins Ross J. Martin Award
CS Students Are Sweet on 3D Printing
Illinois Team Advances to ICPC World Finals
Thieme Shares Her Passion for Technology and for Women in Technology at WCS Banquet
HACK ILLINOIS
Judging will begin at 12:30pm.
HackIllinois 2015 Expo
read more
More News
A
Calendar
Twitter
Facebook
LinkedIn
Instagram
CS Store
Contact Us | Contact Webmaster | Site Map
GETTING HERE
CHICAGO 140 MI
BLOOMINGTON 50 MI
CHAMPAIGN-URBANA
INDIANAPOLIS 120 MI
ST. LOUIS 170 MI

Muhammad Naveed wins Ross J. Martin Award

CS PhD student Muhammad Naveed has been named the winner of the 2016 Ross J. Martin Award by the College of Engineering. Given to only one graduate student across the College each year, the award recognizes outstanding research achievement.

Department of Computer Science
201 North Goodwin Avenue
Urbana, IL 61801-2302

University of Illinois
College of Engineering

©2015 The Board of Trustees at the University of Illinois

Figure 1: In red rectangles we highlight the web lists extracted from a web page taken from *www.cs.illinois.edu*

such as hyperlinks used to enforce the web page authority in a link-based ranking scenario, short-cut hyperlinks, etc. . Moreover, the website structure is codified in navigational systems which provide a local view of the website organization. Navigational systems (e.g. menus, navbars, product lists) are implemented as hyperlink collections having same domain name and sharing layout and presentation properties. In this respect, the solution we propose, based on the usage of web lists, has a twofold effect: from one side it guarantees that only urls useful to the clustering process are considered; on the other side, it allows the method to implicitly take into account the web page structure which is implicitly codified in the web lists available web pages [4], [8], [28], [29]. The crawling algorithm is described in Algorithm 1. In particular, starting from the homepage h, the method *extractWebLists()* is iteratively applied to extract url collections having same domain of h and organized in *web lists*. Only web pages included in web lists are further explored. Following [29], a web list is defined as follows:

Definition 1. A **Web List** is a collection of two or more web elements having similar HTML structure, visually adjacent and aligned on a rendered web page. This alignment can be identified on the basis of the x-axis (i.e. a vertical list), the y-axis (i.e. horizontal list), or in a tiled manner (i.e., aligned vertically and horizontally).

Algorithm 1 crawlingWebsite(homepage)

```
Input: URL homepage;
Output: Set<(URL, URL)> E; Set<(URL, String)> V;
frontier = Set()
Q = Queue(homepage)
repeat
    currentPage = Q. dequeue();
    text = currentPage. getText();
    V. add((currentPage, text));
    webLists = extractWebLists(currentPage);
    for each list $\in$ webLists do
        pagesToAnalyze = list. filterNot(page $\rightarrow$ frontier. contains(page) );
        Q. enqueue(pagesToAnalyze);
        frontier. add(pagesToAnalyze);
        for each u $\in$ pagesToAnalyze do
            E. add((currentPage, u));
        end for
    end for
until !queue.empty()
return (V, E)
```

Fig.1 shows, in red boxes, web lists extracted from the homepage of a computer science department which will be used to guide the website crawler. Links in box A will be excluded because their domains are different from the homepage's domain.

To identify from a web page the set of web lists we implement HyLien [33]. The output of website crawling step is the sub-graph $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, which will be used for link and content vectors generation steps.

3.2. Link vectors generation through Random Walks

A random walk over a linked structure is based on the idea that the connections between nodes encode information about their correlations. Then, the effective semantic of any node in a graph is obtained by analyzing how it is correlated to all other nodes. To capture and codify correlations among graph's nodes (i.e. web pages), which can be indirect, we use the Random Walk with Restart (RWR) approach.

RWR is a Markov chain describing the sequence of nodes (i.e. web pages) visited by a random walker: starting from a random point i, with probability $(1 - \alpha)$ a walker stochastically walks to a new, connected neighbor node or, with probability α, it restarts his walk from i. Algorithm 2 describes the generation process.

In order to use the topology of the graph for capturing correlations among nodes, the generated random walks must be sufficiently long to encode those information. However, on the other side, it is advisable to avoid the effect described in [34], that is, when the length of a random walk starting at vertex i tends towards infinity, the probability of being on a vertex j does not depend on the starting vertex i. In our solution, the length of the random walk, $rwrLength$,

Algorithm 2 rwrGeneration(rwrLength, dbLength, G, α)

```
Input: int rwrLength, int dbLength, Graph G, float $\alpha$;
Output: List<List<URL>> randomWalks;
for each i $\in$ Range(0, dbLength) do
    w = List()
    w[0] = G. getRandomVertex();
    for each j $\in$ Range(1, rwrLength) do
        $\lambda$ = Math.random()
        if $\lambda > \alpha$ then
            w[j] = G. getRandomOutlink(w[j-1]);
        else
            w[j] = w[0]
        end if
    end for
    randomWalks. add(w);
end for
return randomWalks
```

is defined by the user (following indications provided in in [34], in the experiments, we set $rwrLength = 10$).

Inspired by the information retrieval universe, we see a web page as a word, that is, a topic indicator and each random walk as a document constituting the natural context of words (i.e. topical unity). Then, continuing the information-retrieval metaphor, we can represent a collection of random walks as a document collection where topics intertwine and overlap. The idea is to apply any NLP or information retrieval algorithm which uses the distributional hypothesis on document's objects to extract new knowledge [35].

In our case, we apply a state-of-art algorithm, the skip-gram model [36] to extract a vector space representations of web pages that encode the topological structure of the website. In the skip-gram model we are given a word w in a corpus of words V_W (in our case a web page w belonging to random walks) and their contexts $c \in V_C$ (in our case web pages in random walks which appear before and after the web page w).

We consider the conditional probabilities $p(c|w)$, and given a random walks collection Rws generated by Algorithm 2, the goal is to set the parameters θ of $p(c|w;\theta)$ so to maximize the following probability:

$$\underset{\theta}{\operatorname{argmax}} \prod_{L \in Rws; w \in L} \left[\prod_{c \in C_L(w)} prox_L(w,c) \cdot p(c|w;\theta) \right] \tag{1}$$

where L is a random walk in Rws, w is a web page in L and $C_L(w) = \{w_{i-k}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+k}\}$ is the set of contexts of web page w in the list L. Moreover, $prox_L(w,c)$ represents the proximity between w and $c \in C_L(w)$. This is necessary since the skip-gram model gives more importance to the nearby context words than distant context words.

One approach for parameterizing the skip-gram model follows the neural-network language models literature, and models the conditional probability $p(c|w;\theta)$ using soft-max:

$$p(c|w;\theta) = \frac{e^{v_c \cdot v_w}}{\sum_{c' \in V_C} e^{v_{c'} \cdot v_w}} \tag{2}$$

where v_c, $v_{c'}$ and $v_w \in \mathbb{R}^d$ are vector representations for c, c' and w respectively (d is defined by the user). Therefore, the optimization problem (1) leads to the identification of the web page and context matrices $W = \{v_{w_i} | w_i \in V_W\}$ and $C = \{v_{c_i} | c_i \in V_C\}$. They are dependent each other and we only use W to represent web pages (coherently with what proposed in [36] for words).

Equation 2 is computationally expensive due the summation $\sum_{c' \in V_C}$. One way of making the computation more tractable is to replace the softmax with an *hierarchical softmax*. It uses a binary tree representation where words (web pages in our case) are leaves and each node stores the relative probabilities of its child nodes. Using this representation it is possible to evaluate only $\sim log_2(V_C)$ nodes rather than V_C [36].

An alternative approach to hierarchical softmax is represented by the softmax negative-sampling approach (SGNS). In this case the objective function can be formalized as follows:

$$\underset{\theta}{\operatorname{argmax}} \prod_{(w,c) \in D} p((w,c) \in D|c,w;\theta) \prod_{(w,c) \notin D'} p((w,c) \in D|c,w;\theta) \tag{3}$$

where:

- D is the set of all web pages and context pairs we extract from the web site;
- D' is the set of random pairs, assuming they are not present in the web site (i.e. negative examples). In [36] authors suggest a number of negative examples of 5-20 pairs (for each w) for small corpus data and 2-5 pairs (for each w) for big corpus data;
- $p((w,c) \in D|c,w;\theta) = \frac{1}{1+e^{-v_c \cdot v_w}}$ is the probability that the pair (w,c) comes from the corpus data;
- $p((w,c) \notin D|c,w;\theta) = 1 - p(D=1|c,w;\theta)$ is the probability that the pair (w,c) does not come from the corpus data;

Therefore, given in input to skip-gram model a corpus data composed by the collection of random walks, it returns the matrix W which embeds each web page into a dense and low-dimensional space $\mathbb{R}^d$.

3.3. Content vectors generation

In this section we describe the process for generating a vector representation of web pages using textual information. On the contrary of from traditional documents, web pages are written in HTML and contain additional information, such as HTML tags, hyperlinks and anchor text or other than textual content visible in a web browser. To apply on web pages a bag-of-words representation we need to compute a preprocessing step, in which the following operations are performed:

- Remove HTML tags. However, we maintain terms in anchor, title and metadata since they contribute to better organize web pages [37]
- Unescape escaped characters;
- Eliminate non-alphanumeric characters;

- Eliminate too frequent ($> 90\%$) and infrequent ($< 5\%$) words;

After preprocessing, each web page is converted in a plain textual document and we can apply the traditional *TF-IDF* weighting schema to obtain a content-vector representation. Due the uncontrolled and heterogeneous nature of web page contents, vector representation of web pages based on content is characterized by high-dimensional sparse data. To obtain a dense and low-dimensional space we apply Truncated SVD algorithm, a low-rank matrix approximation based on random sampling [38]. In particular, given the *TF-IDF matrix* of size $|V'| \times n$ and the desired dimensionality of content vectors m, where $m << n$, the algorithm returns a denser and lower-dimensional matrix of size $|V'| \times m$.

3.4. Content-link coupled Clustering

Once the content vector $v_c \in \mathbb{R}^m$ and the link vector $v_l \in \mathbb{R}^d$ of each web page in V' have been generated, the last step of the algorithm is to concatenate them in a new vector having dimension $m + d$. Before the concatenation step we normalize each vector with its Euclidean norm. In this way we ensure that components of v_l having highest weights are as important as components of v_c having highest weights.

The matrix generated by concatenation step preserves both structural and textual information and can be used in traditional clustering algorithms based on vector space model. In this study we consider K-MEANS and H-DBSCAN [39] because they are well known and present several complementary properties (distance vs. density-based).

4. Experiments

In order to empirically evaluate our approach, we performed experiments on several real websites. Specifically, we used four computer science department's websites: *Illinois* (cs.illinois.edu), *Princeton* (cs.princeton.edu), *Oxford* (www.cs.ox.ac.ou), *Stanford* (cs.stanford.edu). The motivation behind this choice is related to our competence in manually labelling pages belonging to this domain. This was necessary in order to create a ground truth for the evaluation of the clustering results.

This evaluation has been performed in order to answer to specific research questions: 1) Which is the real contribution of combining content and hyperlink structure in a single vector space representation with respect to using only either textual content or hyperlink structure? 2) Which is the real contribution of exploiting web pages structure (i.e. HTML formatting) and, specifically, the role of using web lists to reduce noise and improve clustering results?

In Table 1 the dimension of each dataset is described. In particular, to correctly analyze the contribution of web lists in the clustering process, we compare only the web pages extracted both by crawling websites using web lists and by traditional crawling (first column of Table 1). Moreover, we report the dimension of the edge set obtained with traditional crawling (second column) and crawling using web lists (third column). Finally the last column describes the number of clusters manually identified by the experts.

TABLE 1: Description of Websites

Website	#pages	#edges	#edges using web lists	#clusters
Illinois	563	9415	5330	10
Oxford	3480	44526	35148	19
Stanford	167	12372	30087	10
Princeton	3132	122493	104585	16

We evaluated the effectiveness of the proposed approach by using the following measures:

- Homogeneity [40]: each cluster should contain only data points that are members of a single class. This measure is computed by calculating the conditional entropy of the class distribution given the proposed clustering. It is bounded below by 0.0 and above by 1.0 (higher is better).
- Completeness [40]: all of the data points that are members of a given class should be elements of the same cluster. Symmetrically to Homogeneity it is computed by the conditional entropy of the proposed cluster distribution given the real class. It ranges between 0 and 1 (higher is better).
- V-Measure [40]: harmonic mean between homogeneity and completeness.
- Adjusted Mutual Information (AMI): it is a variation of the Mutual Information MI. $MI = \sum_{i \in K} \sum_{j \in C} log \frac{P(i,j)}{P(i)P(j))}$ where C is the set of real classes, K is the set of learned clusters, $P(i,j)$ denotes the probability that a point belongs to both the real class i and the learned cluster j and $P(i)$ is the a priori probability that a point falls into i. However MI is generally higher for two clusterings with a larger number of clusters, regardless of whether there is actually more information shared. The Adjusted Mutual Information represents an adjustment of this metric to overcome this limitation.
- Adjusted Random Index (ARI) [41]: it represents a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings. $RI = (a + b)/\binom{n}{2}$ where a is number of pairs of points that are in the same class and learned cluster and b is number of pairs of points that belong to different class and learned cluster. As in the case of AMI, the ARI is an adjustment which ensures to have a value close to 0.0 for random labeling independently of the number of clusters and samples and exactly 1.0 when the clusterings are identical.
- Silhouette: it measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

To response to our research questions we ran our algorithm with different configurations depending on the crawling type (with or without web list constrains) and web pages' information used for the vectors generation. We enumerate the used configurations as follows:

- *Text*. We generate a vector space representation, having dimension $m = 120$, using only web pages' textual information;
- *RW-List*. We generate a vector space representation of size $d = 120$ using only hyperlink structure extracted by crawling the website using web lists. For this scope we ran the *rwrGeneration* algorithm (see Section 3.2) setting the input parameters to $\alpha = 1$, $rwrLength = 10$ and $dbLength = 100k$;
- *RW-NoList*. We generate a vector space representation of size 120 using only the hyperlink structure obtained with traditional crawling. We ran *rwrGeneration* with the same parameters of RW-List;
- *Comb-Lists*. We combine, as defined in Section 3.4, the content vector of size $m = 60$ and hyperlink structure vector of size $d = 60$ generated by crawling the website using web lists. For link vector generation we use the same parameters of other configurations.
- *Comb-NoLists*. As in the Comb-Lists configuration we combine textual and hyperlink structure of web pages in a single vector space representation having size 120. However, on the contrary of Comb-Lists, we use traditional crawling.

Since our goal is not that of comparing clustering algorithms, we set for K-MEANS the parameter K (i.e. total number of clusters to generate) to the number of real clusters, while we set for H-DBSCAN the *minimal cluster size* parameter to 5. Finally, since at the best of our knowledge there is no work which uses the skip-gram model to analyze the topological structure of websites, we ran both of skip-gram versions (i.e hierarchical softmax and SGNS) for generating link vectors. However, due to space limitations, we report only results for SGNS (setting the window size to 5), which, in most cases, outperformed hierarchical softmax.

Tables 2, 3, 4 and 5 present the main results. In general, the experiments show that best results are obtained combining textual information with hyperlink structure. This is more evident for Illinois and Oxford websites, where content and hyperlinks structure codify complementary information for clustering purpose. However, for the Stanford website using the textual information decreases the clustering performance. The importance of combining content and hyperlink structure is confirmed by Nemenyi post-hoc test (see Figure 2) and Wilcoxon signed Rank test (see Table 6). This behaviour is quite uniform for all the evaluation measures considered (see Table 6).

For the last research question, results do not show a statistical contribution in the use of web lists for clustering purpose (see Figure 2 and Table 6). This can be motivated by the fact that analyzed websites are very well structured and poor of noisy links. This can be observed in Table 1, where there is not a big difference in terms of edges number between the real web graph and that one extracted using web lists. However, as expected the Completeness is higher for Comb-Lists, confirming that clusters have higher ”precision” in the case of crawling based on web lists (see Figure 2 b).

5. Conclusions and Future Works

In this paper, we have presented a new method which combines information about content, web page structure and hyperlink structure in a single vector space representation which can be used by any traditional and best-performing clustering algorithms. To take into account the hyperlink structure we exploit recent advances in natural language processing by adapting the skip-gram model. In the evaluation we have analyzed two research questions: 1) Which is the real contribution of combining content and hyperlink structure in a single vector space representation with respect to using only either textual content or hyperlink structure? 2) Which is the real contribution of exploiting web pages structure (i.e. HTML formatting) and, specifically, the role of using web lists to reduce noise and improve clustering results? Experiments results show that content and hyperlink structure of web pages provide different and complementary information which can improve the efficacy of clustering algorithms. Moreover, experiments do not show statistical differences between results which use web lists and results obtained ignoring web page structure. As feature work we will run our algorithm on different domains and less structured websites in the way to observe whether web lists are really useless in the web page clustering process.

References

[1] O. Zamir and O. Etzioni, “Web document clustering: A feasibility demonstration,” in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’98. New York, NY, USA: ACM, 1998, pp. 46–54.

[2] M. H. Chehreghani, H. Abolhassani, and M. H. Chehreghani, “Improving density-based methods for hierarchical clustering of web pages,” *Data Knowl. Eng.*, vol. 67, no. 1, pp. 30–50, Oct. 2008.

[3] T. H. Haveliwala, A. Gionis, D. Klein, and P. Indyk, “Evaluating strategies for similarity search on the web,” in *Proceedings of the 11th International Conference on World Wide Web*, ser. WWW ’02. New York, NY, USA: ACM, 2002, pp. 432–442.

[4] V. Crescenzi, P. Merialdo, and P. Missier, “Clustering web pages based on their structure,” *Data Knowl. Eng.*, vol. 54, no. 3, pp. 279–299, Sep. 2005.

[5] D. Buttler, “A short survey of document structure similarity algorithms,” in *Proceedings of the International Conference on Internet Computing, IC ’04, Las Vegas, Nevada, USA, June 21-24, 2004, Volume 1*, 2004, pp. 3–9.

[6] P. Bohunsky and W. Gatterbauer, “Visual structure-based web page clustering and retrieval,” in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW ’10. New York, NY, USA: ACM, 2010, pp. 1067–1068.

[7] C. X. Lin, Y. Yu, J. Han, and B. Liu, “Hierarchical web-page clustering via in-page and cross-page link structures,” in *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, ser. PAKDD’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 222–229.

[8] X. Qi and B. D. Davison, "Knowing a web page by the company it keeps," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ser. CIKM '06. New York, NY, USA: ACM, 2006, pp. 228–237.

[9] X. He, H. Zha, C. H.Q. Ding, and H. D. Simon, "Web document clustering using hyperlink structures," *Comput. Stat. Data Anal.*, vol. 41, no. 1, pp. 19–45, Nov. 2002.

[10] D. S. Modha and W. S. Spangler, "Clustering hypertext with applications to web searching," in *Proceedings of the Eleventh ACM on Hypertext and Hypermedia*, ser. HYPERTEXT '00. New York, NY, USA: ACM, 2000, pp. 143–152.

[11] Y. Wang and M. Kitsuregawa, "Evaluating contents-link coupled web page clustering for web search results," in *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ser. CIKM '02. New York, NY, USA: ACM, 2002, pp. 499–506.

[12] I. Drost, S. Bickel, and T. Scheffer, "Discovering communities in linked data by multi-view clustering." in *GfKl*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nrnberger, and W. Gaul, Eds. Springer, 2005, pp. 342–349.

[13] R. Angelova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ser. CIKM '06. New York, NY, USA: ACM, 2006, pp. 778–779.

[14] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake, "Using web structure for classifying and describing web pages," in *Proceedings of the 11th International Conference on World Wide Web*, ser. WWW '02. New York, NY, USA: ACM, 2002, pp. 562–569.

[15] M. Halkidi, B. Nguyen, I. Varlamis, and M. Vazirgiannis, "Thesus: Organizing web document collections based on link semantics," *The VLDB Journal*, vol. 12, no. 4, pp. 320–332, Nov. 2003.

[16] P. Calado, M. Cristo, E. Moura, N. Ziviani, B. Ribeiro-Neto, and M. A. Gonçalves, "Combining link-based and content-based methods for web document classification," in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, ser. CIKM '03. New York, NY, USA: ACM, 2003, pp. 394–401.

[17] S. Zhu, K. Yu, Y. Chi, and Y. Gong, "Combining content and link for classification using matrix factorization," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '07. New York, NY, USA: ACM, 2007, pp. 487–494.

[18] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Int. Res.*, vol. 37, no. 1, pp. 141–188, Jan. 2010.

[19] J. Firth, "A synopsis of linguistic theory 1930-1955," *Studies in linguistic analysis*, pp. 1–32, 1957.

[20] O. Gornerup, D. Gillblad, and T. Vasiloudis, "Knowing an object by the company it keeps: A domain-agnostic scheme for similarity discovery," in *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, ser. ICDM '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 121–130.

[21] B. S. Anami, R. S. Wadawadagi, and V. B. Pagi, "Machine learning techniques in web content mining: A comparative analysis," *Journal of Information & Knowledge Management (JIKM)*, vol. 13, no. 01, 2014.

[22] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM Comput. Surv.*, vol. 41, no. 2, pp. 12:1–12:31, Feb. 2009.

[23] S. Helmer, N. Augsten, and M. Böhlen, "Measuring structural similarity of semistructured data based on information-theoretic approaches," *The VLDB Journal*, vol. 21, no. 5, pp. 677–702, Oct. 2012.

[24] M. Cristo, P. Calado, E. S. de Moura, N. Ziviani, and B. A. Ribeiro-Neto, "Link information as a similarity measure in web classification," in *String Processing and Information Retrieval, 10th International Symposium, SPIRE 2003, Manaus, Brazil, October 8-10, 2003, Proceedings*, 2003, pp. 43–55.

[25] M. M. Kessler, "Bibliographic coupling between scientific papers," *American Documentation*, vol. 14, pp. 10–25, 1963.

[26] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *Journal of the American Society for Information Science*, vol. 24, no. 4, pp. 265–269, 1973.

[27] M. Fisher and R. Everson, "When are links useful? experiments in text classification," in *Proceedings of the 25th European Conference on IR Research*, ser. ECIR'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 41–56.

[28] T. Weninger, T. J. Johnston, and J. Han, "The parallel path framework for entity discovery on the web," *ACM Trans. Web*, vol. 7, no. 3, pp. 16:1–16:29, Sep. 2013.

[29] P. F. Lanotte, F. Fumarola, M. Ceci, A. Scarpino, M. Torelli, and D. Malerba, "Automatic extraction of logical web lists," in *Foundations of Intelligent Systems*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8502, pp. 365–374.

[30] Y. Zhou, H. Cheng, and J. X. Yu, "Clustering large attributed graphs: An efficient incremental approach," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, ser. ICDM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 689–698.

[31] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. New York, NY, USA: ACM, 2015, pp. 1067–1077.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 701–710.

[33] F. Fumarola, T. Weninger, R. Barber, D. Malerba, and J. Han, "Hylien: a hybrid approach to general list extraction on the web," in *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, 2011, pp. 35–36.

[34] P. Pons and M. Latapy, "Computing communities in large networks using random walks (long version)," *ArXiv Physics e-prints*, Dec. 2005.

[35] M. Sahlgren, "The distributional hypothesis," *Italian Journal of Linguistics*, vol. 20, no. 1, pp. 33–54, 2008.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 3111–3119.

[37] M. Fathi, N. Adly, and M. Nagi, "Web documents classification using text, anchor, title and metadata information," in *Proceedings of the international conference on computer science, software engineering, information technology, e-Business and Applications*, 2004, pp. 1–8.

[38] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, May 2011.

[39] R. J. G. B. Campello, Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," *Advances in Knowledge Discovery and Data Mining*.

[40] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in NLP and Computational Natural Language Learning(EMNLP-CoNLL)*, 2007, pp. 410–420.

[41] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

TABLE 2: Experimental result for Illinois's website

Configuation	Skip-gram model	Clustering	Homogeneity	Completeness	V-Measure	ARI	AMI	Silhouette
Text	-	KMEANS	0.84	0.62	0.71	0.4	0.61	0.33
Text	-	H-DBSCAN	0.72	0.53	0.61	0.4	0.5	0.21
RW-Lists	SGNS	KMEANS	0.72	0.53	0.61	0.27	0.51	0.42
RW-Lists	SGNS	H-DBSCAN	0.81	0.47	0.6	0.18	0.43	**0.43**
RW-NoLists	SGNS	KMEANS	0.71	0.52	0.6	0.25	0.5	0.42
RW-NoLists	SGNS	H-DBSCAN	0.8	0.45	0.58	0.17	0.41	0.42
Comb-Lists	SGNS	KMEANS	**0.9**	**0.69**	**0.78**	**0.54**	**0.68**	0.4
Comb-Lists	SGNS	H-DBSCAN	0.83	0.51	0.63	0.27	0.48	0.34
Comb-NoLists	SGNS	KMEANS	0.84	0.62	0.71	0.37	0.6	0.38
Comb-NoLists	SGNS	H-DBSCAN	0.83	0.52	0.64	0.27	0.49	0.29

TABLE 3: Experimental results for the Princeton's website

Configuation	Skip-gram model	Clustering	Homogeneity	Completeness	V-Measure	ARI	AMI	Silhouette
Text	-	KMEANS	0.71	**0.59**	**0.64**	**0.68**	**0.58**	0.21
Text	-	H-DBSCAN	0.36	0.31	0.34	0.12	0.28	-0.21
RW-Lists	SGNS	KMEANS	0.56	0.37	0.45	0.27	0.36	0.18
RW-Lists	SGNS	H-DBSCAN	0.49	0.3	0.37	0.12	0.26	-0.05
RW-NoLists	SGNS	KMEANS	0.55	0.36	0.43	0.24	0.35	0.15
RW-NoLists	SGNS	H-DBSCAN	0.48	0.3	0.37	0.1	0.26	-0.09
Comb-Lists	SGNS	KMEANS	0.76	0.54	0.63	0.55	0.53	0.14
Comb-Lists	SGNS	H-DBSCAN	0.47	0.52	0.49	0.36	0.45	**0.37**
Comb-NoLists	SGNS	KMEANS	**0.78**	0.54	**0.64**	0.49	0.53	0.13
Comb-NoLists	SGNS	H-DBSCAN	0.47	0.52	0.49	0.37	0.45	0.38

TABLE 4: Experimental results for the Oxford's website

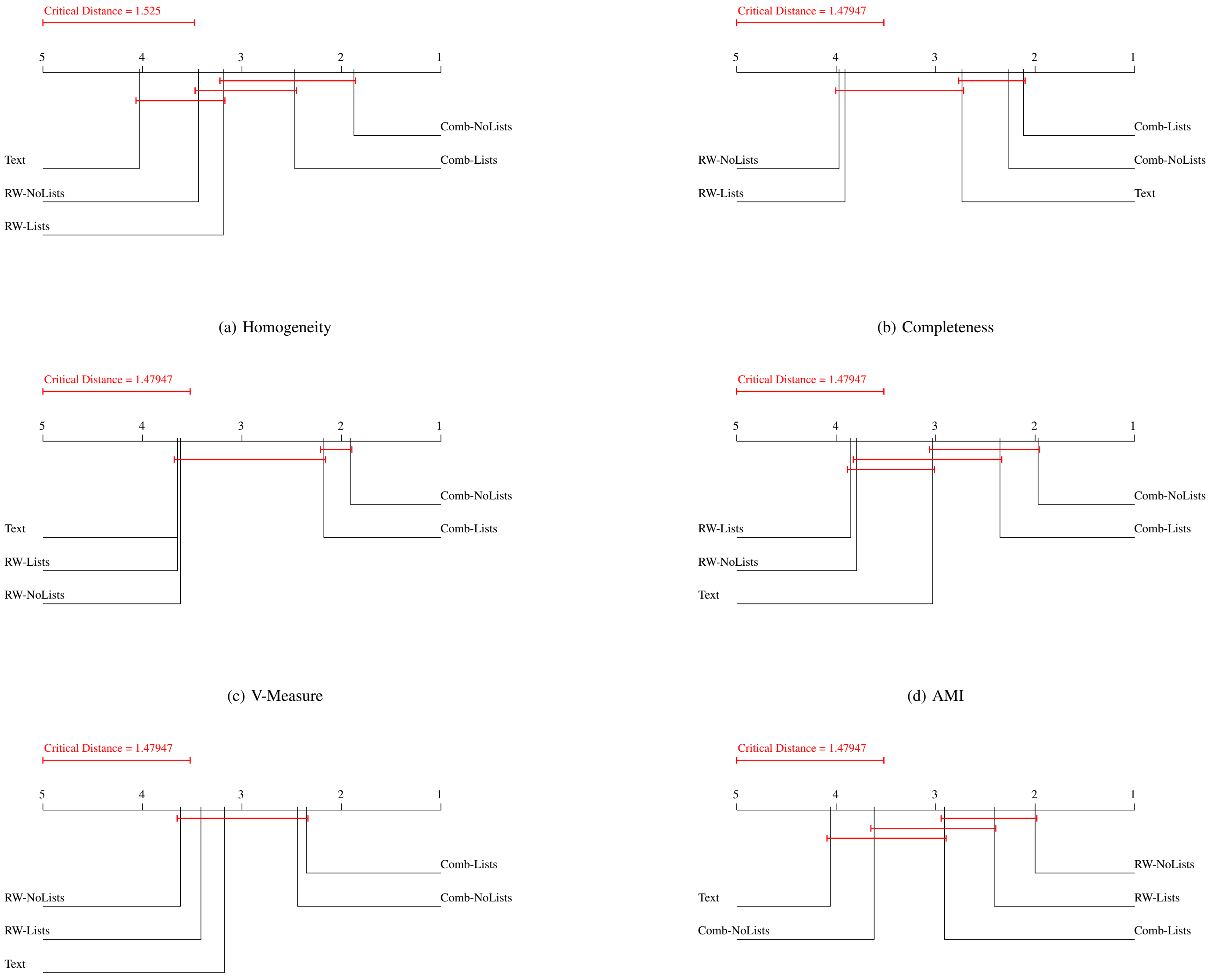
Configuation	Skip-gram model	Clustering	Homogeneity	Completeness	V-Measure	ARI	AMI	Silhouette
Text	-	KMEANS	0.74	0.6	0.66	0.48	0.59	0.25
Text	-	H-DBSCAN	0.43	0.41	0.42	0.07	0.37	-0.06
RW-Lists	SGNS	KMEANS	0.65	0.55	0.6	0.48	0.54	0.32
RW-Lists	SGNS	H-DBSCAN	0.6	0.44	0.51	0.26	0.41	0.22
RW-NoLists	SGNS	KMEANS	0.67	0.57	0.62	0.51	0.56	**0.35**
RW-NoLists	SGNS	H-DBSCAN	0.6	0.45	0.51	0.27	0.41	0.18
Comb-Lists	SGNS	KMEANS	0.79	0.67	0.73	**0.56**	0.67	0.34
Comb-Lists	SGNS	H-DBSCAN	0.58	0.49	0.53	0.15	0.47	0.08
Comb-NoLists	SGNS	KMEANS	**0.81**	**0.68**	**0.74**	0.53	**0.68**	0.28
Comb-NoLists	SGNS	H-DBSCAN	0.62	0.53	0.57	0.23	0.51	0.08

TABLE 5: Experimental results for the Stanford's website

Configuation	Skip-gram model	Clustering	Homogeneity	Completeness	V-Measure	ARI	AMI	Silhouette
Text	-	KMEANS	0.37	0.43	0.39	0.08	0.28	0.3
Text	-	H-DBSCAN	0.18	0.62	0.28	0.07	0.16	0.43
RW-Lists	SGNS	KMEANS	**0.59**	0.58	**0.58**	**0.27**	**0.52**	0.31
RW-Lists	SGNS	H-DBSCAN	0.28	0.4	0.33	0.1	0.22	0.15
RW-NoLists	SGNS	KMEANS	0.47	0.54	0.5	0.14	0.39	0.53
RW-NoLists	SGNS	H-DBSCAN	0.34	0.6	0.43	0.13	0.29	**0.55**
Comb-Lists	SGNS	KMEANS	0.42	0.46	0.44	0.12	0.34	0.22
Comb-Lists	SGNS	H-DBSCAN	0.21	**0.63**	0.31	0.07	0.17	0.46
Comb-NoLists	SGNS	KMEANS	0.53	0.56	0.54	0.17	0.46	0.35
Comb-NoLists	SGNS	H-DBSCAN	0.34	0.51	0.4	0.12	0.28	0.27

TABLE 6: Wilcoxon pairwise signed Rank tests. (+) indicates that the second model wins. (-) indicates that the first model wins. The results are highlighted in bold if the difference is statistically significant (at p-value=0.05). The tests have been performed by considering the results obtained with both hierarchical softmax and SGNS skip-gram models.

	Homogeneity	Completeness	V-Measure	Adj Rand index	Adj Mutual info	Silhouette
Text vs Comb	**(+) 0.000**	(-) 0.055	**(+) 0.000**	(+) 0.342	**(+) 0.003**	**(+) 0.020**
RW vs Comb	**(+) 0.002**	**(+) 0.000**	**(+) 0.000**	**(+) 0.000**	**(+) 0.000**	(+) 0.229
NoLists vs Lists	(-) 0.342	(-) 0.970	(-) 0.418	(+) 0.659	(+) 0.358	(-) 0.362



(a) Homogeneity

(b) Completeness

(c) V-Measure

(d) AMI

(e) ARI

(f) Silhouette

Figure 2: Results of the Nemenyi post-hoc test for the results in terms of Homogeneity, Completeness, V-Measure, AMI, ARI, Silhouette. Better algorithms are positioned on the right-hand side, and those that do not significantly differ in performance (at p-value=0.05) are connected with a line. The tests have been performed by considering the results obtained with both hierarchical softmax and SGNS skip-gram models.